



IT214 (Sec-B)

DATABASE MANAGEMENT SYSTEM

LAB 4 TO 10 REPORT (Version 2 - Final)

Team Id: 5.16

Sahil Mangukiya: 202001251

Manan Modi: 202001254

18.11.2022

Group 5

Sec 5

Case Study:

Hackathon management system

Included:

Final SRS, Final ER Diagram and Noun analysis, Final Normalization and DDL Scripts, SQL Queries, Final Front-end related documents

Subject and Index:

Lab-4.

- Understand & Complete the Description of the Case Study/Problem Domain.
 1. Purpose
 2. Intended Audience and Reading Suggestions
 3. Product Scope
 4. Description

Lab-5.

- Improve the Submitted Final SRS.
 1. Reading and Description
 2. Interview
 3. Questionnaires
 4. Observations
 5. Fact Finding Chart
 6. List Requirements
 7. User Categories and Privileges
 8. List of privileges/functions that can be accessed by different User classes
 9. Assumptions and Business Constraints

Lab-6.

- Perform Noun analysis on Description.
- Develop an ER Diagram based on Noun Analysis.
 1. Complete SRS Description
 2. Noun Analysis
 3. Entity-Attribute
 4. Rejected Nouns
 5. Rejected Verb
 6. ER Diagram of Version 1
 7. ER Diagram of Version 2

Lab-7.

- Finalize the ER Diagram.
 1. Final ER Diagram
 2. Identify

Lab-8.

- Complete ERD final version.
- Convert ERD to Relational Model.
- Create DDL Scripts.
 1. Final Relational Model Diagram
 2. Convert ER Diagram to Relational Model (All Relation)
 3. Final ER Diagram
 4. Functional Dependencies
 5. Redundancy and Analysis

Lab-9.

- Normalization and Schema Refinement
- Re-write DDL Scripts.
 1. Normalization up-to 3NF/BCNF
 2. DDL Scripts
 3. Snapshot of 13 Table
 4. 20 SQL Queries including all Join operation, operator, clause, Trigger, Function, View and Nested Queries.

Lab-10.

- DB connectivity with front end
 1. Front-end
 2. Code and Preview of Home page
 3. Sort, Insert, Update and delete operation on Participants Database
 4. Sort, Insert, Update and delete operation on Hackathon Database
 5. Run Query.

Lab 4

Purpose:

A hackathon's goal is to develop functional software or hardware before the end of the competition. There are various goals for hackathons, however the main purposes of these events are listed below.

- Help Identify Employee Potential - Key Purpose of the Hackathon
- Team-work
- Purpose of the hackathon management system is to provide a platform to conduct online events.
- This system also provides facilities creators to took tests and also monitor them without interrupting the participants.
- This system helps participants to track their progress through this platform and form creators to track the progress of each participant.
- Event manager can see the numbers of registrations happening to participants.
- For participants, these systems help make an efficient team.
- Communication between creators and participants is the key purpose of this kind of system.

Intended Audience and Reading Suggestions:

When planning an event like such a hackathon, consider your target Participants. It's critical to understand what your participants are thinking, accepting, and comfortable with.

After conducting an advance survey that includes everything from the site to the rules, you will be able to understand your target audience. You can learn more about your participants by conducting a survey, which will help you to collect information about them. There is Two types of Hackathons listed here.

Internal hackathon: The target audience for internal hackathons is employees of that company. Both companies and employees benefit from pushing staff to think creatively.

External hackathon: Every student as well as members of the community participating in this hackathon with their specific advanced skills.

<https://www.hackerearth.com/challenges/>, hacker rank etc.

Product Scope:

- The scope is to provide a reliable and concurrent management system that can handle multiple users (participants as well as origination who is conducting).
- The second scope is to give participants the best platform to improve their skills and ensure that cheating does not happen.
- Make a system for organizers to maintain the info regarding the participants efficiently.
- This system also provides facility to participants to go through the code written by other users to optimize the solutions.
- Using this system, you can make a different view for organizers and participants to ensure that the events happen without any partiality.
- Famous products and apps born at hackathons.

Description:

We have to create the first **table** in our system that **organizes table for those who organize the hackathon event**. Now, what is the idea behind it?

First of all, who can organize a hackathon event.

So, the answer is quite simple: High Schools, Universities, Colleges Student Community skill with their advanced skills (like in our college there is a programming club whose student community of advanced programming skills organizes ICPC Round, Code Mutants, etc.). Additionally, Many Multinational companies namely Google organize hackathons events like Google Jam code, kick start, etc.

So, we must create a table for those companies, colleges which organize the hackathon events. Additionally, we must add table attributes (columns). So which

type of Column we need to add namely What is the name of companies, colleges which organize hackathon events? For companies There are more columns we must add such as location, when the company was founded, Number of departments and in each department Number of employees etc.

Second Question is why does any company organize the hackathon event.

So, the answer is placement/Full time offer for those participants who have ability which companies require.

Now, adding the **second table** is all the details about the **type of placement given by the company**. In this table, we have to put placement data of the companies, namely companies package, what type of study (B.Tech, M.Tech, etc.) and profiles (software engineer, database developer, Full stack developer, etc.) they want. Additionally, CPI criteria, Number of Round, company is open for Job or Summer internship or both.

Third question is that Company declares the placement details but what about the participants.

So, this question is the idea behind adding a **third table that participants details**.

Now, what kind of data we have to put in this table is that participants Full name, where is he/she from, namely country, college, etc. If participants are from College, then what is the student id, college email address, CPI, study all these things we have to implement in our participants table.

Now, Company also knows about the participants' resume (meaning what kind of ability, skills participants have).

So, this is the key why we add a fourth **table** that is **participant resume** in our system. According to participants' resumes, we have to implement the table in such a way that all participants' achievements, advanced skills, CPI, experience,

position, interest, Preferred Programming language and tools etc. so, these are all contained in the resume table.

Now, coming to the company, take a test to choose among all participants to those participants who have scored well in the test.

So, companies must declare what kind of questions are in the test and which platform companies use like hackerearth, hackerrank, etc. Then, date and time of the round, duration of the round, Number of Question in round, difficulties of the question (Easy, Medium, Hard). What kind of CS subject is included? All kinds of this data are in this table.

So, **the fifth table** is the **company test round**.

Now, coming to the company test round, there are many rounds to get placement in particular companies like the first round is coding round, second is Group discussion round, third is technical round, fourth is HR round and also add is that all these rounds which are in online mode or offline mode. So, all kinds of data in the **sixth table** is that **round name**.

This is all about the description that we need to implement in the hackathon management system.

Now after getting selected from the coding round the participants must go through the Technical round and the hr round which includes the company name itself, information about the student which is taken from the participants details, the date and time of the round, the duration(how long it would take) of that round, also we have to add that the particular round is conducting in which mode is it offline mode or is it online mode and the most important is the result that the particular company is selecting that particular candidate or not.

So, the **Seventh table** is **technical round + HR Round combine**

This is all about the description that we need to implement in the hackathon management system.

Lab 5

[2]. Document the Requirements Collection/Fact-Finding Phase

2.1 Reading and Description

By going through the following references and performing background reading on them, we took a brief understanding of what the Hackathon data models look like and what functionalities they provide to the different classes of users.

1) Google Summer of Code

Reference: <https://summerofcode.withgoogle.com/>

2) How to run a successful Hackathon

Reference: <https://hackathon.guide/>

From the above two websites, we examined an already existing Hackathon database management system. We referred to the services and functionalities that the database system should offer and maintain. We also inferred the functions and roles that we need to design and maintain and also examined the overall relationship between them.

In simpler terms, we referred to the general structure of the database. From these websites, we understood the overall requirements of the Hackathon database management system and the points to be considered for the smooth functioning of the system and the satisfaction of the users by focusing on their needs.

3) Database System Concepts, by Abraham Silberschatz, Henry F. Korth, S. Sudarshan

Reference: <https://db-book.com/>

From this article, we got familiar with the various DBMS concepts like E-R model, E-R diagrams, Relational model, Mapping of E-R model to Relational model, Database designing and Database development, etc.

Combined requirements gathered from the Background readings:

- A well-functioning Database management system is required to maintain and update all the information.
- A user interface is required so that the users can easily access the data whenever required, without having to know much about the actual implementation of the system.
- System administrators and users should be assigned different roles to ensure the integrity and consistency of the database.
- System structure and functions should be designed in such a way that it ensures the efficient performance of the system.
- The interface should be clean and without any redundant data.

2.2 Interview(s)

[1]. Interview plan and summary

System: Hackathon Management System

Interviewee: Smit Mangukiya (**Role Play**)

Designation: Work at Google

Interviewer:

1. Sahil Mangukiya
2. Manan Modi

Designation: Database Developer

Designation: Database Manager

Date: 30/09/2022

Time: 15:00

Duration: 45 mins

Place: Google Meet

Purpose of interview:

Preliminary meeting to identify problems and requirements of Organizer regarding the hackathon database management system.

Agenda:

- Requirements analysis of the interviewee
- Initial ideas
- Follow-up actions

Summary of Results and Requirements:

- Designing/choosing a theme and how it can be linked to the principles and image of the company that is offering the event.
- Definition of goals and objectives of the hackathon.
- Organize software, and hardware type of hackathon.
- Duration of the event.

[2]. Interview plan and summary

System: Hackathon Management system

Interviewee: Manan Diyora (**Role Play**)

Designation: Work at Microsoft

Interviewer:

1. Sahil Mangukiya
2. Manan Modi

Designation: Database Developer

Designation: Database Manager

Date: 01/10/2022

Time: 15:00

Duration: 45 mins

Place: Google meet

Purpose of interview:

Preliminary meeting to identify problems and requirements of Organizer regarding the hackathon database management system.

Agenda:

- Requirements analysis of the athletes
- Initial ideas
- Follow-up actions

Summary of Results and Requirements:

- Publicizing the Event
- Announcement and invitation to encourage participation
- Hackathon Timing - When will the hackathon be held
- Social media and Team building for the Mentor and like-minded people.

[3]. Interview plan and summary

System: Hackathons Management system

Interviewee: Sunny Dhameliya (**Role Play**)

Designation: Work at Tekion

Interviewer:

1. Sahil Mangukiya
2. Manan Modi

Designation: Database Developer
Designation: Database Manager

Date: 02/10/2022

Time: 15:00

Duration: 45 mins

Place: Zoom

Purpose of interview:

Preliminary meeting to identify problems and requirements of Organizer regarding the hackathon database management system.

Agenda:

- Requirements analysis of the athletes
- Initial ideas
- Follow-up actions

Summary of Results and Requirements:

- Running Opening ceremony
- Website for Registering in hackathon
- Mentorship
- Connecting to Mentors to participants

[4]. Interview plan and summary

System: Hackathon Management System

Interviewee: Harshil Soni (**Role Play**)

Designation: Work at wells Fargo

Interviewer:

- 3. Sahil Mangukiya
- 4. Manan Modi

Designation: Database Developer

Designation: Database Manager

Date: 03/10/2022

Time: 15:00

Duration: 30 mins

Place: Zoom

Purpose of interview:

Preliminary meeting to identify problems and requirements of Organizer regarding the hackathon database management system.

Agenda:

- Requirements analysis of the interviewee
- Initial ideas
- Follow-up actions

Summary of Results and Requirements:

- Communication Platform to communicate with participants namely discord and slack are most popular.
- Registration and send Reminders
- Fundraising, Digital sponsorship
- Prize distribution

[5]. Interview plan and summary

System: Hackathon Management System

Interviewee: Om Thummer (**Role Play**)

Designation: Work at Tekion

Interviewer:

- 5. Sahil Mangukiya
- 6. Manan Modi

Designation: Database Developer

Designation: Database Manager

Date: 04/10/2022

Time: 15:00

Duration: 45 mins

Place: Google meet

Purpose of interview:

Preliminary meeting to identify problems and requirements of Organizer regarding the hackathon database management system.

Agenda:

- Requirements analysis of the interviewee
- Initial ideas
- Follow-up actions

Summary of Results and Requirements:

- Streaming/Video Conferencing Platform for digital events.
- Placeholder Website, which includes event date, pre-registration form, support email, social handling, release date, sponsor interest page, etc.
- Promoting hackathons events.

[6]. Interview plan and summary

System: Hackathon Management System

Interviewee: Kashyap panchani(**Role Play**)

Designation: Convener of programming club

Interviewer:

- 7. Sahil Mangukiya
- 8. Manan Modi

Designation: Database Developer
Designation: Database Manager

Date: 05/10/2022

Time: 15:00

Duration: 30 mins

Place: Discord

Purpose of interview:

Preliminary meeting to identify problems and requirements of Organizer regarding the hackathon database management system.

Agenda:

- Requirements analysis of the interviewee
- Initial ideas
- Follow-up actions

Summary of Results and Requirements:

- Hackathon Budgeting
- Award ceremony
- Choosing Target audience
- Hackathon prizes
- Judge and speakers

Combined requirements gathered from the Interviews:

➤ Finding the purpose

As you begin to plan your hackathon, it's important to think about the purpose of your event. Brainstorm with your organizing team the why behind your event.

➤ Finding the Date

After you've solidified the purpose of your event, it's time to pick a date! Like Major holidays and festivals, University breaks, etc.

➤ Theme Poster of hackathons

To create excitement and increase the number of hackathons' participation by making the event public.

➤ Place holder website

Until an event has a website, it doesn't exist for most people.

➤ Understanding your sponsor

Sponsorship is one of the most crucial parts of every hackathon. When you're reaching out to sponsors, it's necessary to understand their perspective of supporting your hackathon and the outcome they're expecting out of it.

➤ Hackathon budgeting

Budgets are the main starting point in determining how much your sponsorship requirements for the hackathon will be. It also helps determine the logistical load your organizing team is looking to take on and which tools and services can help you do your best and most productive work economically.

➤ **Promoting hackathon event**

Marketing your hackathon is another one of the major areas of focus one has while planning it out. The aim here is to reach as many people as possible with the focus on the right/Target audience/community meant for the event.

➤ **Registration on the hackathon website**

Once all the above procedures are done, the organizer launched the hackathon's official website to start the registration procedure.

➤ **Mentor recruitment**

To facilitate the process, set up a webpage mentioning a mentor's responsibilities and the perks they might be getting by getting enrolled as a mentor for your hackathon. Set up a form to get interested people to register, asking them to pick a timeslot to help facilitate mentorship.

➤ **Communication platform**

Pick a communication platform and ensure participants join it. Slack and Discord are two of the most popular.

➤ **Video conferencing platform**

To carry out the communication more effectively, the organizer can also allow face-to-face meetings among mentors and participants on various video conferencing platforms namely google meet, Webex, zoom, discord video call, etc.

➤ **Sending reminders**

Finally, a week before your hackathon, it is the perfect time to remind everyone involved about what needs to happen at the event.

➤ **Prize distribution/ Award ceremony**

Once the duration of the hackathon event is over, Now, it's time to distribute the prizes based on the result obtained by evaluation of the project made by a different team.

2.3 Questionnaires

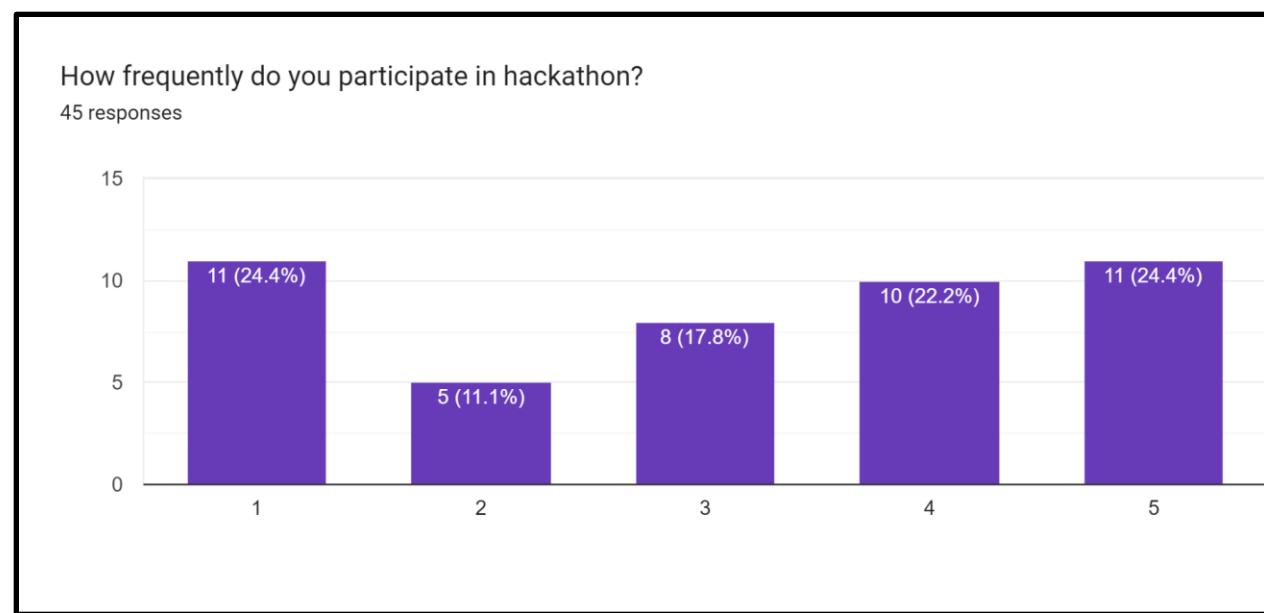
Questions were asked/surveyed via a Google form, and the summary of the responses is shown below:

➤ How frequently do you participate in hackathons?

How frequently do you participate in hackathon?

1 2 3 4 5

Occassionally Regularly



- **Intent of the question:**

To get an idea about how frequently the users will access the Hackathons database system.

- **Observation from the response:**

The majority of responses indicated that such a database would be beneficial and valuable to all types of audiences.

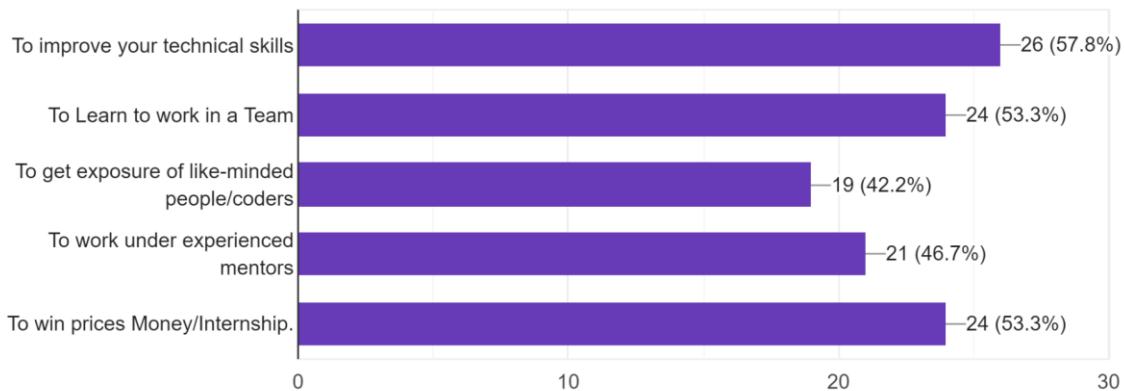
➤ Why do you prefer to give hackathons?

Why do you prefer to give Hackathon? *

- To improve your technical skills
- To Learn to work in a Team
- To get exposure of like-minded people/coders
- To work under experienced mentors
- To win prizes Money/Internship.

Why do you prefer to give Hackathon?

45 responses



● Intent of the question:

To determine the participant's purpose for taking part in the hackathon.

● Observation from the response:

More significant data regarding the technical skills participants obtain from various hackathons should be included in the database.

➤ What Type of Hackathons Would You Like to Participate in?

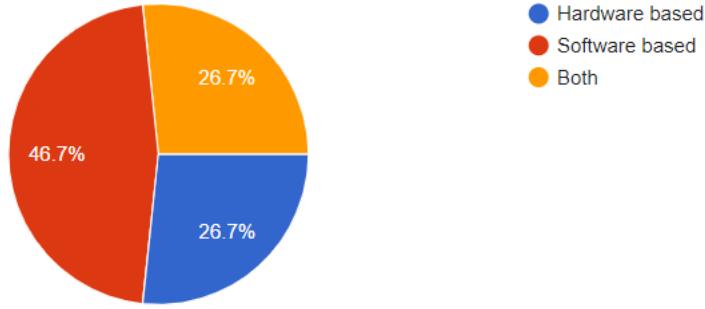
What Type of Hackathons Would you like to participate in? *

- Hardware based
- Software based
- Both

What Type of Hackathons Would you like to participate in?

 Copy

45 responses



- Intent of the question:

To determine what types of hackathon data my database should have more of.

- Observation from the response:

Participants get more information about software implementation-based hackathon.

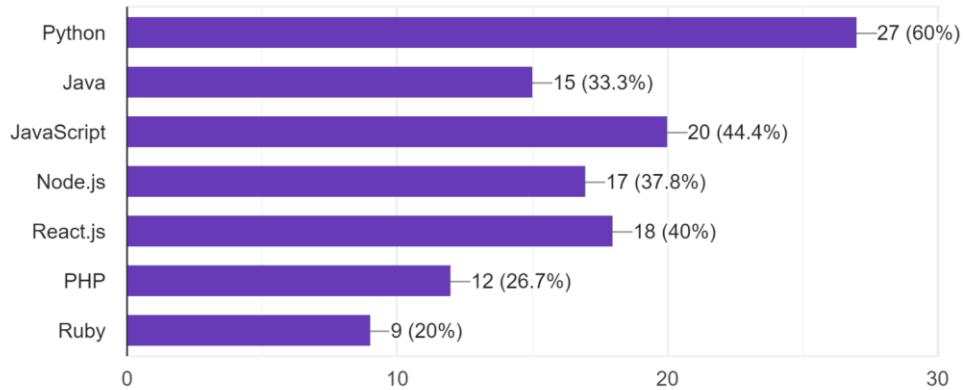
➤ What language do you prefer in Hackathons?

What language do you prefer in a Hackathons? *

- Python
- Java
- JavaScript
- Node.js
- React.js
- PHP
- Ruby

What language do you prefer in a Hackathons?

45 responses



- **Intent of the question:**

To know which language are more use in software based hackathons.

- **Observation from the response:**

participants are more comfortable with python and javascript-based software hackathon.

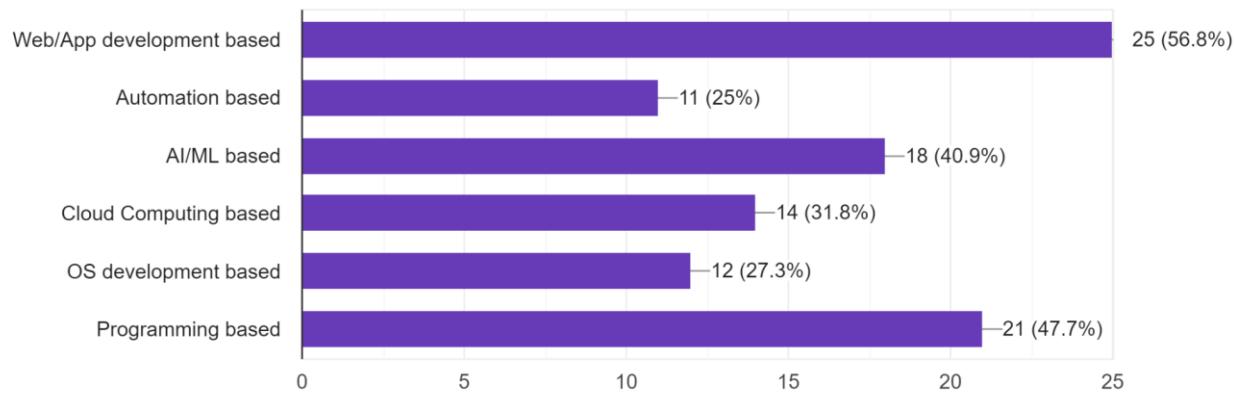
➤ What type of Project domain do you like in hackathons?

What type of Project domain do you like in hackathons?

- Web/App development based
- Automation based
- AI/ML based
- Cloud Computing based
- OS development based
- Programming based

What type of Project domain do you like in hackathons?

44 responses



● Intent of the question:

To know which domains is more use in software-based hackathons.

● Observation from the response:

participants are more comfortable with web/app development and programming-based software hackathon.

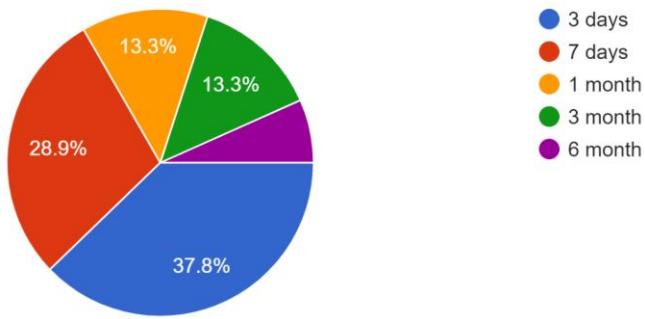
➤ What duration of hackathons do you prefer?

Whay duration of hackathons do you prefer?

- 3 days
- 7 days
- 1 month
- 3 month
- 6 month

Whay duration of hackathons do you prefer?

45 responses



- **Intent of the question:**

To know what duration is more used in software-based hackathons.

- **Observation from the response:**

We can see in the above chart that many participants are interested in short-run hackathons rather than long-run hackathons.

Combined requirements gathered from the Questionnaire:

- The database system to be developed should have a good UI and should provide efficient services.
- Regular updates of the data stored in the database system.
- Database should more information about short time hackathon.
- Database should contained pythons and javascript language based project.
- Web/App development and programming domain should be included in the hackathon.
- User feedback regularly.
- Incorporation of the additional requirements proposed by the users during the feedback.

2.4 Observations

System: Hackathon Management system

Observations by: Sahil Mangukiya, Manan Modi

Date: 04/10/2022

Time: 17:00

Duration: 45 minutes

Place: Google Summer of Code [LINK](#)

Observations and Combined Requirements:

- Google Summer of Code's major projects, events, and students going to develop are all covered in the most recent news.
- From the perspective of a contributor, every project and organization information is clearly indicated.
- Concurrent access to the database system.
- Well-designed user interface that should withstand traffic up to a certain level and should provide a quality experience to the users.
- Efficient services related to the search, insert, delete, and update operations.
- Real-time update of all the information stored in the database system, maintaining consistency of the database.
- Efficient and optimal storage of all data in the system.

[3]. Fact Finding Chart

Collecting data from examining documents, interviewing, questionnaires, observations, and research.

Objective	Technique	Subject	Time commitment
To gain background knowledge about hackathon systems and events	Background readings	Few relatable officials and third-party apps/websites	1 day
To gain an understanding of the roles of Hackathon organizer	Interview	Google's Employee	45 mins
To gain an understanding of the roles of Hackathon organizer	Interview	Microsoft Employee	45 mins
To gain an understanding of the roles of Hackathon organizer	Interview	Convener of programming club	45 mins
To gain an understanding of the user's perspective	Questionnaire	General Public	1 day
To gain an understanding of the real world hackathon database	Observation	Google summer of code website (Similar site)	1 hour

[4]. List Requirements

- Date and purpose of hackathon event. (**got at background reading, interview**)
- Build Hackathon leadership Team. (**interview**)
- Theme poster for hackathon event. (**Questionnaire, interview**)
- Look down for a venue. (**interview, observation**)
- Set up for placeholder website. (**interview, Questionnaire, background reading**)
- Understanding between hackathon organizers and sponsors who give sponsorship. (**interview**)
- Determine Hackathon budgeting. (**background reading, observation**)
- Good UI and efficient service of database. (**observation, background reading, Questionnaire**)
- Regular update of the store in the database. (**background reading, observation**)
Putting together a sponsorship prospectus. (**interview**)
- Promoting hackathon events. (**Interview, background reading**)
- Make the hackathon website live (in public). (**background reading, interview**)
- Well-design user of the interface. (**observation**)
- Collecting registrations for the hackathon. (**interview, Questionnaire**)
- Used communication software in the hackathon. (**Questionnaire, interview**)
- Start registration of hackathon mentor. (**interview**)
- Efficient Services like delete, update, search, insert, etc. (**observation, Questionnaire**)
- Sending reminders to the participants. (**interview**)
- Award ceremony/prize distribution. (**interview, Questionnaire**)
- User feedback regularly. (**Questionnaire, observation**)

[5]. User Categories and Privileges

List of user categories and their roles:

1. Participants (hackers):

During the hackathon event, participants work with like-minded people and showcase their hidden advanced skills. They can learn team-building skills, Technical skills, learn from each other, and experience mentors.

2. Mentor:

The role of the Mentor is to study or analyze the performance of the student and guide them accordingly to improve their performance in the hackathon event.

3. Organizer:

The role of the Organizer is to decide what kind of language and domain are included in the Hackathon event. Additionally, choose mentors according to their capabilities, and achievements and determine the prize distribution for winning students.

4. Database system developer:

The role of the database system developer is to analyze all the requirements. After requirements analysis, the role of the developer is to decide, design, and develop the overall database of the system.

5. Database system manager:

The role of the database system manager is to look after the technical issues faced by the users, control the traffic over the database system and regularly update the data stored in the database system, maintaining the consistency and integrity of the database.

List of privileges/functions that can be accessed by different user classes

1. Search:

- Participants - Participants can search for what events is going on a site.
- Mentor
- Organizer - After searching organizer can know that what is the current Requirement of the participants.
- Developer - After searching Developer can know that if this organizer what this Specification in her events than developer can help them.
- Manager

2. Insert:

- Organizer - They can insert the questions in given events and they can also add constraints in them.
- Developer
- Manager

3. Delete:

- Organizer - They can insert the questions in given events and they can also add constraints in them.
- Developer - If they found that there is any design issue in that management system then they can delete that function and after than they update it.
- Manager - Manager helps on the technical pop on management system they have a access to delete a particular segment if bug found.

4. Update:

- Organizer
- Developer
- Manager

[6]. Assumptions

- It is assumed that the users of this database system will have all the necessary hardware and software required for running this application.
- Moreover, it is assumed that the information stored in the database gets updated as per real-time.
- It is assumed that the users of this database system have stable internet connection.
- The database is consistent.
- The integrity of the data is maintained throughout the database.
- The users will have access to alternative resources in the event of any unprecedented situation.

[7]. Business Constraints

- The amount of software and hardware required for the database system is limited.
- The amount of computing power available for the database system is also limited.
- The capacity of the database is limited.
- The number of users that can access the database simultaneously is also limited.

NOTE:

During lab5, I was not present due to the re-examination of CT303. So, version 1 was submitted by Manan-202001254, and As the lower id is mine (Sahil-202001251), I have submitted version 2.

Lab 6 & 7

Complete SRS Description:

We have to create the first **table** in our system that **organizes table for those who organize the hackathon event.**

High Schools, Universities, Colleges Student Community skill with their advanced skills (like in our college there is a programming club whose student community of advanced programming skills organizes ICPC Round, Code Mutants, etc.). Additionally, Many Multinational companies namely Google organize hackathons events like Google Jam code, kick start, etc.

So, we must create a table for those companies, colleges which organize the hackathon events. Additionally, we must add table attributes (columns). So which type of Column we need to add namely What is the name of companies, colleges which organize hackathon events? For companies There are more columns we must add such as location, when the company was founded, Number of departments and in each department Number of employees etc.

Second is why does any company organize the hackathon event.

So, the answer is placement/Full time offer for those participants who have ability which companies require.

Now, adding the **second table** is all the details about the **type of placement given by the company**. In this table, we have to put placement data of the companies, namely companies package, what type of study (B.Tech, M.Tech, etc.) and profiles (software engineer, database developer, Full stack developer, etc.) they want. Additionally, CPI criteria, Number of Round, company is open for Job or Summer internship or both.

Third is that Company declares the placement details but what about the participants.

So, this statement is the idea behind adding a **third table** that **participants details.**

Now, what kind of data we have to put in this table is that participants Full name, where is he/she from, namely country, college, etc. If participants are from college, then what is the student id, college email address, CPI, study all these things we have to implement in our participants' table.

Now, Company also knows about the participants' resume (meaning what kind of ability, skills participants have).

So, this is the key why we add a **fourth table** that is **participant resume** in our system. According to participants' resumes, we have to implement the table in such a way that all participants' achievements, advanced skills, CPI, experience, position, interest, Preferred Programming language and tools etc. so, these are all contained in the resume table.

Now, coming to the company, take a test to choose among all participants to those participants who have scored well in the test.

So, companies must declare what kind of questions are in the test and which platform companies use like hackerearth, hackerrank, etc. Then, date and time of the round, duration of the round, Number of Question in round, difficulties of the question (Easy, Medium, Hard). What kind of CS subject is included? All kinds of this data are in this table.

So, **the fifth table** is the **company test round**.

Now, coming to the company test round, there are many rounds to get placement in particular companies like the first round is coding round, second is Group discussion round, third is technical round, fourth is HR round and also add is that all these rounds which are in online mode or offline mode. So, all kinds of data in the **sixth table** are that **round name**.

This is all about the description that we need to implement in the hackathon management system.

Now after getting selected from the coding round the participants must go through the Technical round and the hr round which includes the

company name itself, information about the student which is taken from the participants details, the date and time of the round, the duration(how long it would take) of that round, also we have to add that the particular round is conducting in which mode is it offline mode or is it online mode and the most important is the result that the particular company is selecting that particular candidate or not.

So, the **Seventh table is technical round + HR Round combine**

This is all about the description that we need to implement in the hackathon management system.

Now, move on to Reading and description. We obtained a basic understanding of how the Hackathon data models appear and what features they offer to the various user classes. Google summer of code is one of the best examples of Hackathon.

From Background reading, we gathered requirement is that a well-functioning database management system is required to maintain and update all the information. Second is, the interface should be clean and without any redundant data. System administrators and users should be assigned different roles to ensure the integrity and consistency of database. A user interface is required so that the users can easily access the data whenever required, without having to know much about the actual implementation of the system. System structure and functions should be designed in such a way that it ensures the efficient performance of the system.

The next thing is that is interviews. The main purpose of interview is that what is the requirement of interviewee to organize the hackathon event. So, six interview was taken by me and Manan.

Now, Purpose of all interview is preliminary meeting to identify problems and requirements of organizer regarding the hackathon database management system. Requirements analysis of the interviewee, initial ideas, follow-up action are agenda of interview.

The first interviewee is Smit Mangukiya who want to organize hackathon event for employee of the google company (internal hackathon). The

requirement of interviewee is that Designing/choosing a theme and how it can be linked to the principles and image of the company that is offering the event, Definition of goals and objectives of the hackathon, organize software, and hardware type of hackathon and the main thing is what is the duration of the hackathon event. To create excitement and increase the number of hackathons' participation by making the event public.

The second interview is Manan diyora who want to organize external Hackathon event. The main problem of interviewee is how he publicizing the event. So, solution of this problem is promoting hackathon event. Marketing hackathon is another one of the major areas of focus one has while planning it out. The aim here is to reach as many people as possible with the focus on the right/Target audience/community meant for the event and announcement and invitation to encourage participation. Social media and Team building for the Mentor and like-minded people. To facilitate the process, set up a webpage mentioning a mentor's responsibilities and the perks they might be getting by getting enrolled as a mentor for your hackathon. Set up a form to get interested people to register, asking them to pick a timeslot to help facilitate mentorship.

The Third interviewee is Sunny Dhameliya who want to organize external hackathon event. The main problem of interviewee is that how he connection to mentors and participants during the hackathon event. So, first is communication platform, pick a communication platform and ensure participants join it. Slack and Discord are two of the most popular and second is video conferencing platform, to carry out the communication more effectively, the organizer can also allow face-to-face meetings among mentors and participants on various video conferencing platforms namely google meet, Webex, zoom, discord video call, etc.

The Forth interviewee is Harshil Soni who want to organize external hackathon for those students who are interested in competitive programming. So, requirement of interviewee is placeholder website, until an event has a website, it doesn't exist for most people. After registered in hackathon a week before your hackathon, it is the perfect

time to remind everyone involved about what needs to happen at the event. Once the duration of the hackathon event is over, Now, it's time to distribute the prizes based on the result obtained by evaluation of the project made by a different team.

The fifth interviewee is Om Thummer who wants to organize the internal hackathon event for his company. Finding the purpose and date are the requirement of the interviewee. As you begin to plan your hackathon, it's important to think about the purpose of your event. Brainstorm with your organizing team the why behind your event. After you've solidified the purpose of your event, it's time to pick a date! Like Major holidays and festivals, University breaks, etc.

The sixth interviewee is Kashyap Panchani, Convener of programming club who wants to organize internal hackathon event for only daiict student. Requirement of the interviewee is that how to choose target audience. So, conduct a survey via google form and student choose according to their interested field. Second requirement is hackathon budgeting, Budgets are the main starting point in determining how much your sponsorship requirements for the hackathon will be. It also helps determine the logistical load your organizing team is looking to take on and which tools and services can help you do your best and most productive work economically.

Now, we move on to Questionnaires. The main idea is organizer have to choose target audience. So, Questionnaires are the most important step to organize the hackathon event. To get an idea about how frequently the users will be accessing the Hackathons database system, to determine the participants' purpose for taking part in the hackathon, to determine what types of hackathon data my database should have more of, to know which language is more used in software-based hackathons, to know what duration are more use in software-based hackathons.

The Combined requirements gathered from the Questionnaire is that the database system to be developed should have a good UI, provide efficient services and regular update. Database should provide more

information about short time hackathons. User feedback regularly. Incorporation of the additional requirements proposed by the users during the feedback.

Now, the next thing is observation of any hackathon website. Google summer of code is one of best hackathon event in world. The site observation is major projects, events, and students going to develop are all covered in the most recent news. From the perspective of a contributor, every project and organization information are clearly indicated. Well-designed user interface that should withstand traffic up to a certain level and should provide a quality experience to the users. Concurrent access, efficient and optimal storage, real time update, maintaining consistency in database.

Fact finding chart is process of collecting data from examining documents, interviewing, questionnaires, observations, and research. Table contain objective of database, time commitment, Technique like interview, Questionnaire, background reading, observation, etc.

The next thing is User Categories and Privileges, what is categories of user and their role. According to our SRS, there are five type of user categories.

Participants: During the hackathon event, participants work with like-minded people and showcase their hidden advanced skills. They can learn team-building skills, technical skills, learn from each other, and experience mentors.

Mentor: The role of the Mentor is to study or analyse the performance of the student and guide them accordingly to improve their performance in the hackathon event.

Organizer: The role of the Organizer is to decide what kind of language and domain are included in the Hackathon event. Additionally, choose mentors according to their capabilities, and achievements and determine the prize distribution for winning students.

Database system developer: The role of the database system developer is to analyse all the requirements. After requirements analysis, the role of the developer is to decide, design, and develop the overall database of the system.

Database system Manager: The role of the database system manager is to look after the technical issues faced by the users, control the traffic over the database system and regularly update the data stored in the database system, maintaining the consistency and integrity of the database.

Functions that can be accessed by different user classes. There are four functions in our hackathon management system. Search is accessed by all but insert, delete and update are accessed by only organizer, developer, and manager.

Now, move on to the main thing Assumptions. So, there are many assumptions in our system. User have all necessary hardware and software to run this application, user of this database system has stable internet connection, integrity and consistency are maintain in database.

The last thing is business constraints. The number of user, capacity, amount of software, hardware and computing power available for the database system are limited.

Noun Analysis:

No.	Noun	Verb
1.	Hackathon	Organize
2.	Events	Work
3.	Employee	Will
4.	Management system	Be
5.	Duration of round	Can
6.	Intended audience	Keep
7.	Reading suggestion	Maintain
8.	Creators	Provide
9.	Participants	Access
10.	Event manager	Allow
11.	System	Using
12.	Purpose	Provide

13.	Number of Project	Have
14.	information	Many
15.	site	Be
16.	Internal hackathon	
17.	External Hackathon	
18.	Company Name	Found
19.	Branch	
20.	staff	Have
21.	Students	Can
22.	First name	Detail
23.	Last name	Detail
24.	Middle name	Detail
25.	Member	Participating
26.	Community	can

27.	Product scope	About
28.	Users	Access
29.	Concurrent management system	Can
30.	organization	Indicate
31.	Platform	Helps
32.	code	write
33.	organizer	Regards
34.	product	Born
35.	Stipend	Gives
36.	application	Born
37.	High School	Organize
38.	Email id	Have
39.	University	Will
40.	Programming club	Skills

41.	Preferred language	Update
42.	Student community	Organize
43.	Target Participants	Participate
44.	Google	Organize
45.	table	add
46.	column	update
47.	attributes	Gives
48.	location	Provide
49.	Number of Department	Count
50.	Numbers of employee	apply
51.	placement	be
52.	Experience	
53.	Profile	
54.	CPI criteria	Need

55.	Time of round	
56.	Database Developer	
57.	CS subject	Preferred
58.	Job	apply
59.	Summer internship	
60.	Name	Request
61.	Student id	Provide
62.	College email	Access
63.	College name	list
64.	Type Open For	
65.	CPI	Range
66.	Subject	Selected
67.	Duration	
68.	Numbers of Questions	

69.	Number of que solved	Has
70.	Date of round	
71.	Software engineer	Can
72.	Data	
73.	Coding round	declares
74.	Profile	upload
75.	HR Round	wait
76.	Round name	
77.	Interview	
78.	CEO	
79.	Category	
80.	Row	
81.	Result	
82.	Mode	can

83.	Technical Round	wait
84.	Hackathon Management system	
85.	Target audience	
86.	company	list
87.	employee	upload
88.	Platform	
89.	System	
90.	Hackathon events	
91.	Company was founded	
92.	Database manager	
93.	Mentor	
94.	Platform	
95.	Concurrent management system	
96.	Solution	

Entity-Attribute:

Sr. No	Candidate Entity Set	Candidate Attribute Set	Candidate Relationship set
1	Participants	<u>Part_Id</u> , First_Name, Last_Name, DOB, Email_add, Student/Employee, College/Company_Name, CPI	Apply
2	Hackathon	<u>Hack_Id</u> , Hack_Name, Hack_Type, Name_Type, Date, Time, Duration	Apply, Award, Stipend, Package
3	Internship	<u>Stipend_given</u> , Mode	Kind, Stipend
4	Full time offer	<u>Part_id</u> , Package_LPA, Location, Criteria	Kind, Package
5	Prize Hackathon	<u>Prize_id</u> , Hack_id, Prize_1, Prize_2, Prize_3	Award, declare, Help
6	Sponsorship	<u>Sponsor_id</u> , Sponsor_Name	Help
7	Round	<u>Platform</u> , No_of_question, Date_of_round, Time_of_Round, difficulty_of_round	Kind
8	Result of event	<u>Hack_id</u> , Part_id, First_Name, Last_Name, Winning Prizes	declare

Rejected Nouns:

Sr. No.	Noun	Reject Reason
1.	Case study	Vague
2.	table	general
3.	Management system	Duplicate
4.	Intended audience	General
5.	Reading suggestion	General
6.	Event manager	Vague
7.	System	General
8.	information	General
9.	Internal hackathon	General
10.	External Hackathon	General
11.	name	Attribute
12.	Purpose	Irrelevant

13.	Member	Irrelevant
14.	Community	General
15.	Users	General
16.	Platform	Vague
17.	code	General
18.	product	General
19.	application	General
20.	High School	Irrelevant
21.	organization	General
22.	University	General
23.	Student community	Association
24.	Google	General
25.	Student id	Attribute
26.	location	Attribute

27.	Target Participants	General
28.	Database Developer	Association
29.	Job	General
30.	Data	General
31.	placement	Irrelevant
32.	Hackathon management system	General
33.	Declare	Association
34.	Experience	Attribute
35.	CPI criteria	Attribute
36.	Summer internship	Vague
37.	solution	general
38.	Mentor	vague
39.	data	General
40.	Target audience	Association

41.	mode	general
42.	duration	association
43.	Concurrency management system	Duplicates
44.	Hackathon events	general
45.	Company was found	association

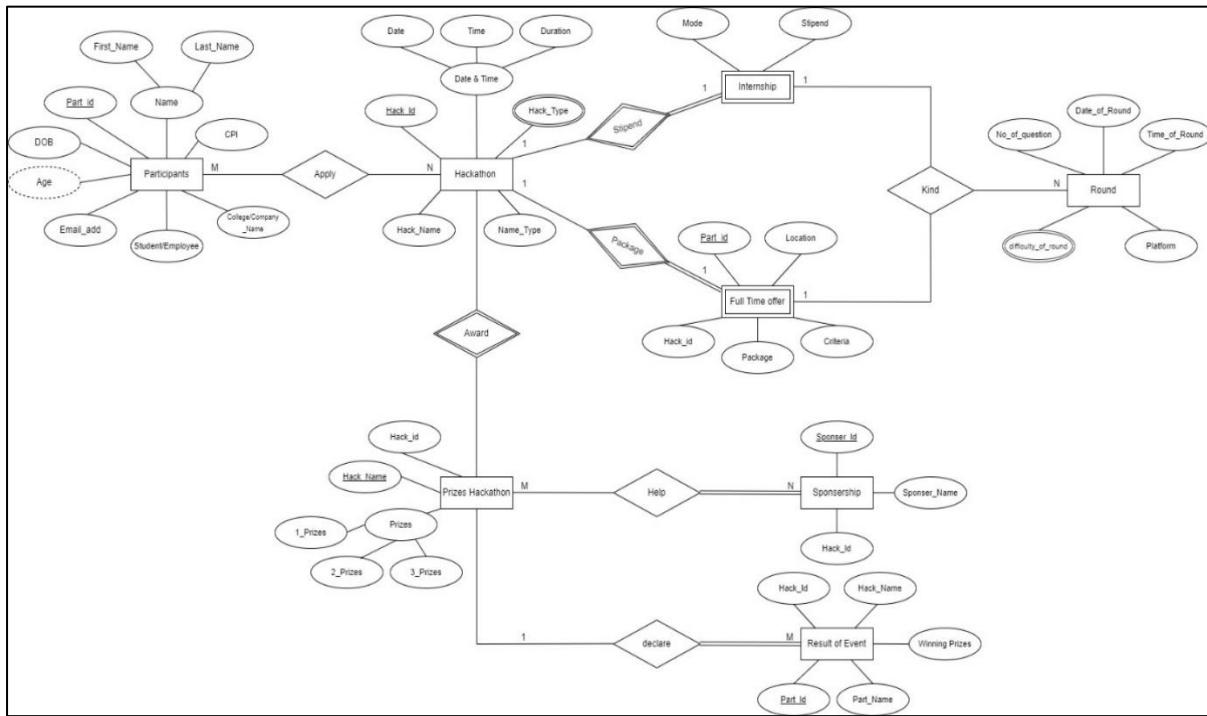
Rejected Verb:

Sr. no	Verb	Reject Reason
1.	Organize	General
2.	Will	General
3.	Can	General
4.	Keep	General
5.	Maintain	General
6.	Allow	General
7.	Provide	General
8.	Be	General
9.	Access	Association
10.	takes	Association

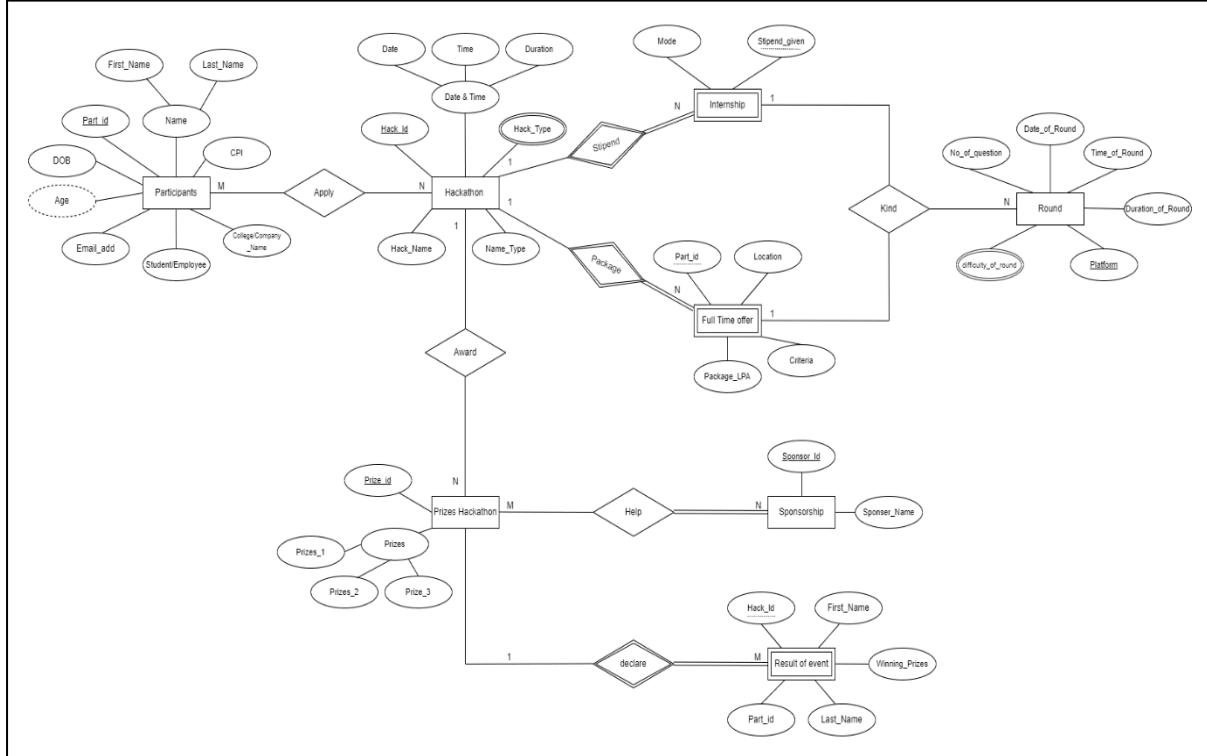
11.	Born	general
12.	Will	general
13.	give	Association
14.	Organize	Vague
15.	provide	General
16.	Take	Association
17.	Found	Vague
18.	Detail	General
19.	upload	Association
20.	Can	General
21.	List	Association

22.	Selected	Vague
23.	work	General
24.	Helps	Vague
25.	Regards	Vague
26.	add	Vague
27.	has	Vague
28.	update	General
29.	provide	Vague
30.	preferred	Vague
31.	apply	Association
32.	about	Vague
33.	take	Association

ER Diagram of Version 1:



ER Diagram of Version 2:



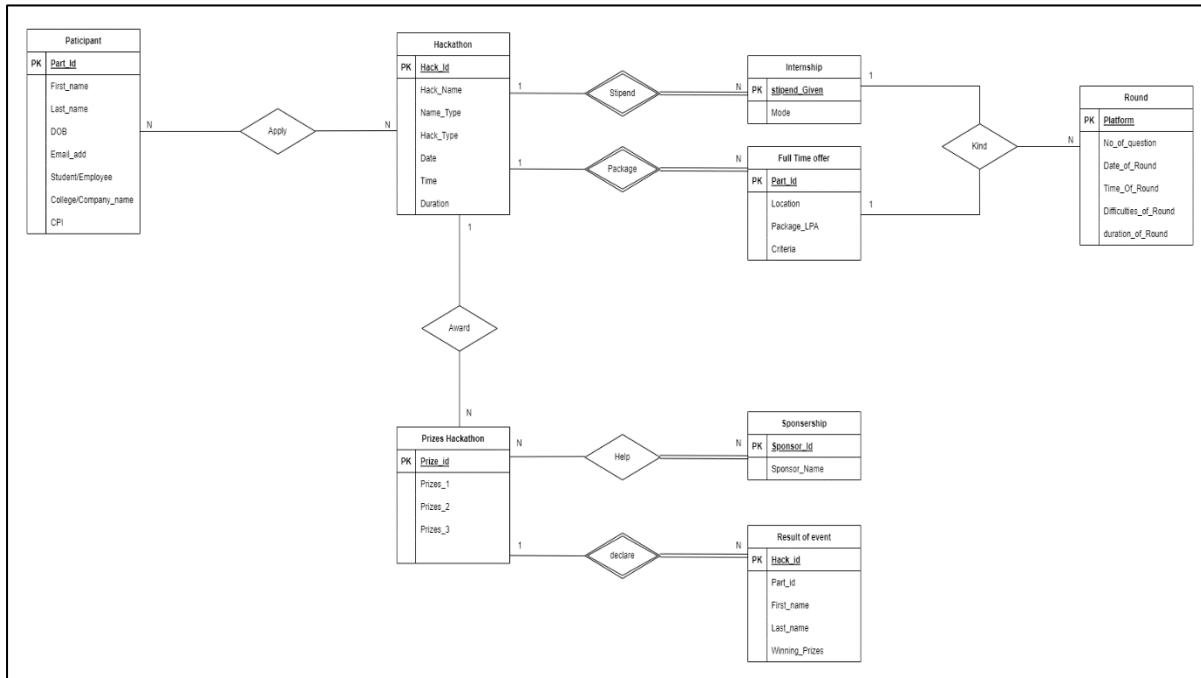
Identify:

Shape	Name
	Attribute
	Multi-valued Attribute
	Derived Attribute
	Primary Key of Entity
	Entity
	Weak Entity
	Relationship
	Identify Relationship
	Total participation

- Here, 1 and N on the relationship denote one to many respectively.
- Here, N and 1 on the relationship denote many to one respectively.
- Here, 1 and 1 on the relationship denote one to one respectively.
- Here, N and N on the relationship denote many to many respectively.
- Here, the double line indicates total participation whereas single line indicates partial participation.
- Double circled attributes denote multi-valued attributes and dotted circled attributes denote derived attributes.
- Here Primary Key (PK) is denoted by underline on an attribute.
- Here dotted underline denotes partial discriminator key for those entities which have composite key.

Derived Attribute	Age (Represent in desh-circle)
Multivalued Attribute	Hack_Type, Difficulties_of_round (represent in double circle)
Composite Attribute	First_Name, Last_Name, Date, Time, Duration, prize_1, prize_2, prize_3
Weak entity	Internship, Full time offer, Result of event
Strong entity	Hackathon, participation, round, Prize Hackathon, Sponsorship
Identify Relationship	Stipend, Package, declare
Relationship	Apply, Award, Kind, Help
Primary Key	Part_id, Hack_id, Prize_id, Platform, Sponsor_id

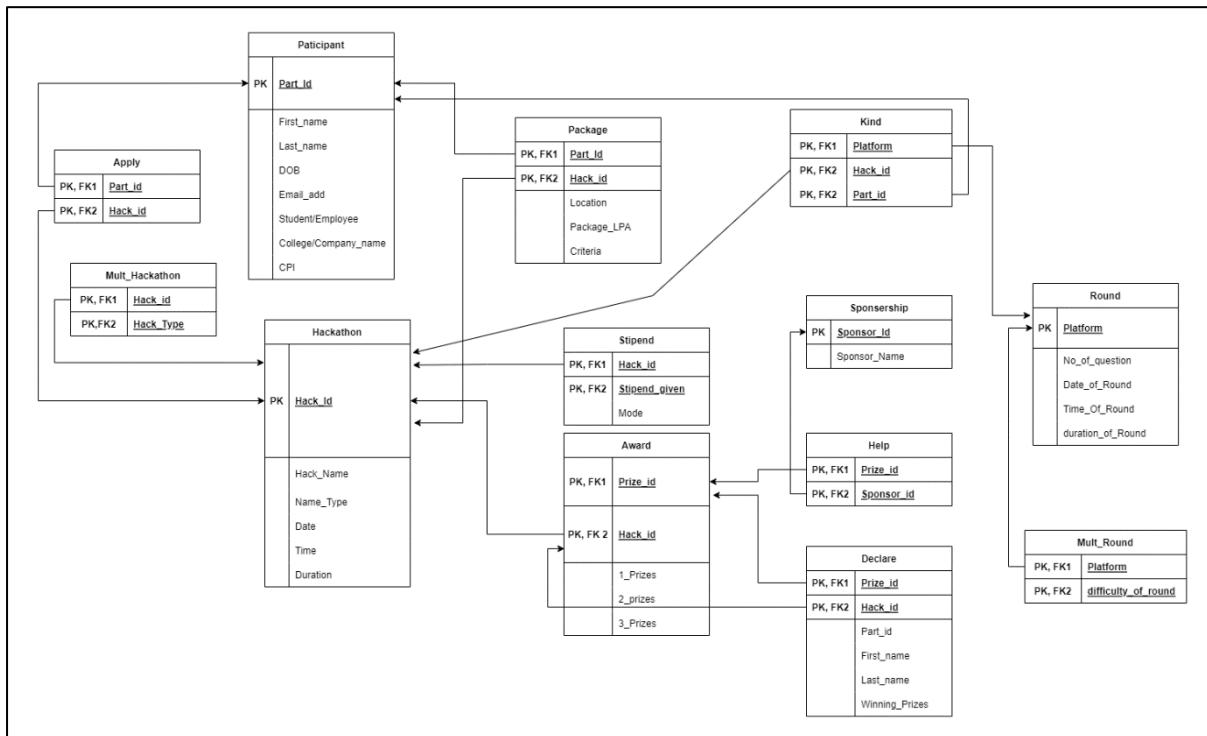
Final ER Diagram:



Here, **Part_id**, **Stipend_given**, **Hack_id** is **discriminator key** for particular table. This is our Final ER-diagram of Hackathon Management System.

Lab 8 & 9

Final Relational Model Diagram:



Convert ER Diagram to Relational Model (All Relation):

- **Participants** (Part_id, First_Name, Last_Name, Email_add, DOB, Student/Employee, College/Company_Name, CPI)
 - FK Part_id references to Participants
 - FK Hack_id references to Hackathon
- **Apply** (Part_id, Hack_id)
 - FK Part_id references to Participants
 - FK Hack_id references to Hackathon
- **Hackthon** (Hack_id, Hack_Name, Name_Type, Date, Time, Duration)
- **Mult_Hackathon** (Hack_id, Hack_Type)
 - FK Hack_id references to Hackathon
- **Stipend** (Hack_id, Stipend_given, Mode)
 - FK Hack_id references to Hackathon
- **Award** (Prize_id)
 - FK Hack_id references to Hackathon

- **Package** (Hack_id, Part_id, Location, Package_LPA, Criteria)
 - FK Hack_id references to Hackathon
 - FK Part_id references to Participants
- **Award** (Prize_id, Hack_id, Prize_1, Prize_2, Prize_3)
 - FK Hack_id references to Hackathon
- **Help** (Prize_id, Sponsor_id)
 - FK Prize_id references to Award
 - FK Sponsor_id references to Sponsorship
- **Sponsorship** (Sponsor_id, Sponsor_Name)
- **Declare** (Prize_id, Hack_id, Part_id, First_Name, Last_Name, Winning_Prizes)
 - FK Prize_id references to Award
 - FK Hack_id references to Award
- **Round** (Platform, No_of_question, date_of_round, time_of_round, duration_of_round)
- **Mult_Round** (Platform, difficulty_of_round)
 - FK Platform references to Round
- **Kind** (Hack_id, Part_id, Platform, Stipend_given)
 - FK Hack_id references to Hackathon
 - FK Part_id references to Participants
 - FK Platform references to Round

Functional Dependencies:

Participants ($\text{Part_id} \rightarrow \text{First_Name}$, $\text{Part_id} \rightarrow \text{Last_Name}$, $\text{Part_id} \rightarrow \text{Email_add}$, $\text{Part_id} \rightarrow \text{DOB}$, $\text{Part_id} \rightarrow \text{Student}$, $\text{Part_id} \rightarrow \text{College}$, $\text{Part_id} \rightarrow \text{CPI}$)

Apply: No Dependencies

Hackathon ($\text{Hack_id} \rightarrow \text{Hack_Name}$, $\text{Hack_id} \rightarrow \text{Name_Type}$, $\text{Hack_id} \rightarrow \text{Date}$, $\text{Hack_id} \rightarrow \text{Time}$, $\text{Hack_id} \rightarrow \text{Duration}$)

Mult_Hackathon: No Dependencies

Stipend ((Hack_id , Stipend_given) \rightarrow Mode)

Package ((Hack_id , Part_id) \rightarrow Location, (Hack_id , Part_id) \rightarrow Package_LPA, (Hack_id , Part_id) \rightarrow Criteria)

Award ($\text{Prize_id} \rightarrow \text{Hack_id}$, $\text{Prize_id} \rightarrow \text{Prize_1}$, $\text{Prize_id} \rightarrow \text{Prize_2}$, $\text{Prize_id} \rightarrow \text{Prize_3}$)

Help: No Dependencies

Sponsorship ($\text{Sponsor_id} \rightarrow \text{Sponsor_Name}$)

Declare ((Hack_id , Prize_id) \rightarrow Part_id, (Hack_id , Prize_id) \rightarrow First_Name, (Hack_id , Prize_id) \rightarrow Last_Name, (Hack_id , Prize_id) \rightarrow Winning_Prizes)

Round ($\text{Platform} \rightarrow \text{No_of_question}$, $\text{Platform} \rightarrow \text{date_of_round}$, $\text{Platform} \rightarrow \text{time_of_round}$, $\text{Platform} \rightarrow \text{duration_of_round}$)

Mult_Round: No Dependencies

Kind: No Dependencies

Redundancy and Analysis:

- **Participants** (Part_id, First_Name, Last_Name, Email_add, DOB, Student, College, CPI)
 - There are no transitive dependencies.
 - There are no anomalies in this entity.
- **Apply** (Part_id, Hack_id)
 - FK Part_id references to Participants
 - FK Hack_id references to Hackathon
 - There are no transitive dependencies.
 - There are no anomalies in this entity.
- **Hackthon** (Hack_id, Hack_Name, Name_Type, Date, Time, Duration)
 - There are no transitive dependencies.
 - There are no anomalies in this entity.
- **Mult_Hackathon** (Hack_id, Hack_Type)
 - FK Hack_id references to Hackathon
 - There are no transitive dependencies.
 - There are no anomalies in this entity.
- **Stipend** (Hack_id, Stipend_given, Mode)
 - FK Hack_id references to Hackathon
 - There are no transitive dependencies.
 - There are no anomalies in this entity.
- **Package** (Hack_id, Part_id, Location, Package_LPA, Criteria)
 - FK Hack_id references to Hackathon
 - FK Part_id references to Participants
 - There are no transitive dependencies.
 - There are no anomalies in this entity.
- **Award** (Prize_id, Hack_id, Prize_1, Prize_2, Prize_3)
 - FK Hack_id references to Hackathon
 - There are no transitive dependencies.
 - There are no anomalies in this entity.

■ Help (Prize_id, Sponsor_id)

- FK Prize_id references to Award
- FK Sponsor_id references to Sponsorship
- There are no transitive dependencies.
- There are no anomalies in this entity.

■ Sponsorship (Sponsor_id, Sponsor_Name)

- There are no transitive dependencies.
- There are no anomalies in this entity.

■ Declare (Prize_id, Hack_id, Part_id, First_Name, Last_Name, Winning_Prizes)

- There are no transitive dependencies.
- There are no anomalies in this entity.
- FK Hack_id references to Award
- FK Prize_id references to Award

■ Round (Platform, No_of_question, date_of_round, time_of_round, duration_of_round)

- There are no transitive dependencies.
- There are no anomalies in this entity.

■ Mult_Round (Platform, difficulty_of_round)

- FK Platform references to Round
- There are no transitive dependencies.
- There are no anomalies in this entity.

■ Kind (Hack_id, Part_id, Platform)

- FK Hack_id references to Hackathon
- FK Part_id references to Participants
- FK Platform references to Round
- There are no transitive dependencies.
- There are no anomalies in this entity.

Normalization up-to 3NF/BCNF:

■ Participants (Part_id, First_Name, Last_Name, Email_add, DOB, Student, College, CPI)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Part_id) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

■ Apply (Part_id, Hack_id)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 candidate key (Part_id, Hack_id). There are no non-prime attributes in the schema. Hence, it is in 2NF form.
- There are no functional dependencies. Hence, it is in 3NF form.

■ Hackthon (Hack_id, Hack_Name, Name_Type, Date, Time, Duration)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Hack_id) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies that it is in 3NF form.

■ Mult_Hackathon (Hack_id, Hack_Type)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 candidate key (Hack_id, Hack_Type). There are no non-prime attributes in the schema. Hence, it is in 2NF form.
- There are no functional dependencies. Hence, it is in 3NF form.

■ Stipend (Hack_id, Stipend_given, Mode)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 candidate key (Hack_id, Stipend_given). There are no non-prime attributes that are dependent on a part of the candidate key. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

■ Package (Hack_id, Part_id, Location, Package_LPA, Criteria)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 candidate key (Hack_id, Part_id). There are no non-prime attributes that are dependent on a part of the candidate key. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

■ Award (Prize_id, Hack_id, Prize_1, Prize_2, Prize_3)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Prize_id) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies that it is in 3NF form.

■ Help (Prize_id, Sponsor_id)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 candidate key (Prize_id, Sponsor_id). There are no non-prime attributes in the schema. Hence, it is in 2NF form.
- There are no functional dependencies. Hence, it is in 3NF form.

■ Sponsorship (Sponsor_id, Sponsor_Name)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Sponsor_id) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies that it is in 3NF form.

■ Declare (Prize_id, Hack_id, Part_id, First_Name, Last_Name, Winning_Prizes)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 candidate key (Prize_id, Hack_id). There are no non-prime attributes that are dependent on a part of the candidate key. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies it is 3NF form.

■ Round (Platform, No_of_question, date_of_round, time_of_round, duration_of_round)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 attribute (Platform) in the candidate key and only one candidate key. There are no partial dependencies. Hence, it is in 2NF form.
- All the functional dependencies have a candidate key on the left side. So, it is in BCNF form which implies that it is in 3NF form.

■ Mult_Round (Platform, difficulty_of_round)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 candidate key (Platform, difficulty_of_round). There are no non-prime attributes in the schema. Hence, it is in 2NF form.
- There are no functional dependencies. Hence, it is in 3NF form.

■ Kind (Hack_id, Part_id, Platform)

- This schema does not have any composite or multivalued attribute. It satisfies atomicity. Hence, it is already in 1NF form.
- There is only 1 candidate key (Hack_id, Part_id, Platform). There are no non-prime attributes in the schema. Hence, it is in 2NF form.
- There are no functional dependencies. Hence, it is in 3NF form.

DDL Scripts:

Create SCHEMA system

Set search_path to system

-----Participants-----

CREATE TABLE IF NOT EXISTS system.Participants

(

 Part_id bigint NOT NULL,
 First_Name varchar NOT NULL,
 Last_Name varchar NOT NULL,
 Email_add char VARYING NOT NULL,
 DOB date NOT NULL,
 Student VARCHAR NOT NULL,
 College VARCHAR NOT NULL,
 CPI FLOAT NOT NULL,
 PRIMARY KEY (Part_id)

)

-----Hackathon-----

CREATE TABLE IF NOT EXISTS system.Hackathon

(

 Hack_id bigint NOT NULL,
 Hack_Name char VARYING NOT NULL,
 Name_Type varchar NOT NULL,
 Date DATE NOT NULL,
 Time TIME NOT NULL,
 Duration bigint NOT NULL,
 PRIMARY KEY (Hack_id)

);

Mult_Hackathon

```
CREATE TABLE IF NOT EXISTS system.Mult_Hackathon
(
    Hack_id bigint NOT NULL,
    Hack_Type varchar NOT NULL,
    FOREIGN KEY (Hack_id) REFERENCES Hackathon(Hack_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid
);
```

Apply

```
CREATE TABLE IF NOT EXISTS system.Apply
(
    Hack_id bigint NOT NULL,
    Part_id bigint NOT NULL,
    FOREIGN KEY (Hack_id) REFERENCES Hackathon(Hack_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid,
    FOREIGN KEY (Part_id) REFERENCES Participants(Part_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid,
    primary key(Hack_id, Part_id)
);
```

Stipend

```
CREATE TABLE IF NOT EXISTS system.Stipend
(
    Hack_id bigint NOT NULL,
    Stipend_given bigint NOT NULL,
    Mode varchar NOT NULL,
    FOREIGN KEY (Hack_id) REFERENCES Hackathon(Hack_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid,
    primary key(Hack_id, Stipend_given)
);
```

Package

```
CREATE TABLE IF NOT EXISTS system.Package
(
    Hack_id bigint NOT NULL, Part_id bigint NOT NULL,
    Location varchar NOT NULL,
    Criteria Float NOT NULL,
    Package_LPA bigint NOT NULL,
    FOREIGN KEY (Hack_id) REFERENCES Hackathon(Hack_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid,
    FOREIGN KEY (Part_id) REFERENCES Participants(Part_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid,
    primary key(Hack_id, Part_id)
);
```

Award

```
CREATE TABLE IF NOT EXISTS system.Award
(
    Prize_id bigint NOT NULL,
    Hack_id bigint NOT NULL,
    Prize_1 bigint NOT NULL,
    Prize_2 bigint NOT NULL,
    Prize_3 bigint NOT NULL,
    PRIMARY KEY (Prize_id),
    FOREIGN KEY (hack_id) REFERENCES Hackathon(Hack_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid
);
```

Sponsorship

```
CREATE TABLE IF NOT EXISTS system.Sponsorship
(
    Sponsor_id bigint NOT NULL,
    Sponsor_Name varchar NOT NULL,
    PRIMARY KEY (Sponsor_id)
);
```

Help

```
CREATE TABLE IF NOT EXISTS system.Help
(
    Prize_id bigint NOT NULL,
    sponsor_id bigint NOT NULL,
    FOREIGN KEY (prize_id) REFERENCES Award(prize_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid,
    FOREIGN KEY (Sponsor_id) REFERENCES Sponsorship(Sponsor_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid
);
```

Declare

```
CREATE TABLE IF NOT EXISTS system.Declare
(
    Hack_id bigint NOT NULL,
    Prize_id bigint NOT NULL,
    Part_id bigint NOT NULL,
    First_Name varchar NOT NULL,
    Last_Name varchar NOT NULL,
    Winning_Prizes bigint NOT NULL,
    FOREIGN KEY (prize_id) REFERENCES Award(prize_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid,
```

```
FOREIGN KEY (Hack_id) REFERENCES Hackathon(Hack_id)
ON Delete Cascade
On Update Cascade
Not Valid
);
```

Round

```
CREATE TABLE IF NOT EXISTS system.Round
(
    Platform varchar NOT NULL,
    No_of_question bigint NOT NULL,
    Date_of_round DATE NOT NULL,
    Time_of_round TIME NOT NULL,
    Duration_of_round bigint NOT NULL,
    PRIMARY KEY (Platform)
);
```

Mult_Round

```
CREATE TABLE IF NOT EXISTS system.Mult_Round
(
    Platform varchar NOT NULL,
    difficulty_of_round varchar NOT NULL,
    FOREIGN KEY (Platform) REFERENCES Round(Platform)
    ON Delete Cascade
    On Update Cascade
    Not Valid
);
```

-----kind-----

```
CREATE TABLE IF NOT EXISTS system.Kind
(
    Part_id bigint NOT NULL,
    Hack_id bigint NOT NULL,
    Platform varchar NOT NULL,
    FOREIGN KEY (Hack_id) REFERENCES Hackathon(Hack_id)
        ON Delete Cascade
        On Update Cascade
        Not valid,
    FOREIGN KEY (Part_id) REFERENCES Participants(Part_id)
        ON Delete Cascade
        On Update Cascade
        Not Valid,
    FOREIGN KEY (Platform) REFERENCES Round(Platform)
        ON Delete Cascade
        On Update Cascade
        Not Valid
);
```

Snapshot of Table:

- Participants (Part_id, First_Name, Last_Name, Email_add, DOB, Student, College, CPI)

Query Query History

```

16
17 COPY participants FROM 'D:\DBMS\final\final\Participation_Table_1.csv' DELIMITER ',' CSV HEADER;
18 select * from participants
19

```

Data output Messages Notifications

	part_id [PK] bigint	first_name character varying	last_name character varying	email_add character varying	dob date	student character varying	college character varying	cpi double precision
30	30	Antoinette	Perigeaux	aperigeauxt@yelp.com	1997-03-16	Employee	Universidad Agra...	9.48
31	31	Godfree	Muriel	gmurielu@artisteer.co...	1992-07-26	Student	Universidad Pont...	5.54
32	32	Katey	Ethridge	kethridgev@artisteer...	2002-05-21	Employee	Bethel College St...	5.09
33	33	Denna	Kelwick	dkelwickw@unesco.org	1999-02-21	Employee	American Univer...	8.94
34	34	Eolande	Earpe	earpex@woothemes....	1993-07-17	Employee	Northeast Norma...	7.04
35	35	Karleen	Beinke	kbeinkey@phpbb.com	1989-06-01	Employee	Kyungpook Natio...	4.23
36	36	Morry	Goodband	mgoodbandz@xrea.c...	1990-10-13	Student	Sarhad University...	9.42
37	37	Buck	Chappelow	bchappelow10@biglo...	1990-11-21	Student	Mohammad Ali J...	8.47
38	38	Mirabel	Bunton	mbunton11@stanford...	1992-01-03	Employee	Boston University	7.44
39	39	Rosalinda	Dinjes	rdinjes12@ehow.com	1991-04-19	Employee	Universidad de Z...	7.47
40	40	Shoshana	Gallelli	sgallelli13@icq.com	2000-07-08	Employee	Universidad Tecn...	6.76
41	41	Lurline	Brackenbury	lbrackenbury14@nsw...	2003-02-28	Employee	Rochester Institu...	8.88
42	42	Vevay	Itzkovwitch	vitzkovwitch15@blog...	1990-02-22	Employee	University of Dayt...	9.17
43	43	Franzen	Sallans	fsallans16@vinaora.c...	1991-09-23	Student	Sacred Heart Uni...	6.84
44	44	Darrel	Student	dStudent17@tinypic.c...	1995-12-12	Student	Northwood Unive...	7.35
45	45	Hermie	Perkins	hperkins18@state.gov	2004-01-05	Student	Northern Arizona...	6.56

Total rows: 45 of 45 Query complete 00:00:00.185

Number of Tuples → 45

■ **Hackthon** (Hack_id, Hack_Name, Name_Type, Date, Time, Duration)

Query History												
33	COPY Hackathon FROM 'D:\DBMS\final\final\Hackathon_1.csv' DELIMITER ',' CSV HEADER;											
34	select * from Hackathon											
35												
36												
Data output Messages Notifications												
      												
	hack_id [PK] bigint	hack_name character varying	name_type character varying	date date	time time without time zone	duration bigint						
30	30	Home Ing	Job	2023-10-22	16:20:00	3						
31	31	It	Internship	2023-03-26	18:09:00	3						
32	32	Alpha	Internship	2023-02-22	15:25:00	4						
33	33	Pannier	Internship	2023-02-25	16:18:00	3						
34	34	Zontrax	Job	2023-09-03	12:16:00	4						
35	35	Gembucket	Job	2023-04-03	18:43:00	3						
36	36	Redhold	Job	2023-11-01	14:58:00	3						
37	37	It	Programming	2022-11-27	15:25:00	2						
38	38	Tempsoft	Job	2023-01-28	15:52:00	4						
39	39	Alphazap	Internship	2023-10-14	14:47:00	2						
40	40	Mat Lam Tam	Programming	2023-02-23	18:57:00	2						
41	41	Cardguard	Job	2023-06-20	13:04:00	4						
42	42	Otcom	Job	2023-03-06	12:44:00	3						
43	43	Rank	Programming	2023-02-08	13:11:00	2						
44	44	Opela	Programming	2023-10-05	17:17:00	2						
45	45	Wrapsafe	Job	2023-11-11	13:59:00	3						
Total rows: 45 of 45 Query complete 00:00:00.085												

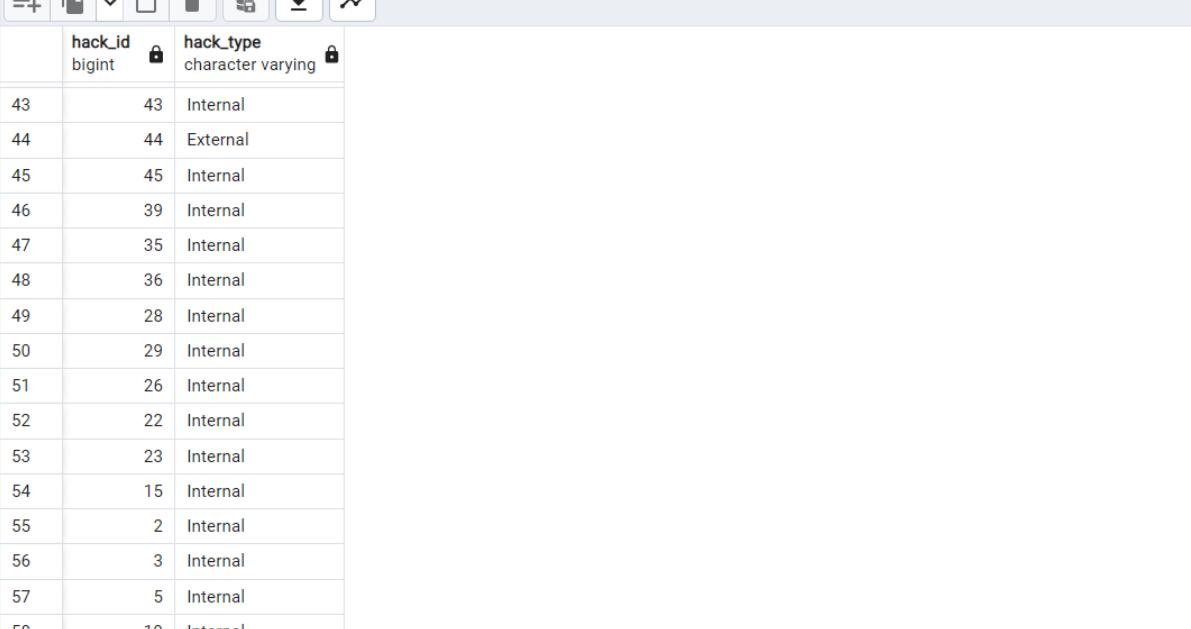
Number of Tuples → 45

■ Mult_Hackathon (Hack_id, Hack_Type)

Query Query History

```
48 COPY Mult_Hackathon FROM 'D:\DBMS\final\final\Mult_Hackathon_1.csv' DELIMITER ',' CSV HEADER;
49 select * from Mult_Hackathon
50
51
```

Data output Messages Notifications

A screenshot of a database query interface showing the results of a SELECT query on the 'Mult_Hackathon' table. The table has two columns: 'hack_id' (datatype bigint) and 'hack_type' (datatype character varying). The data consists of 58 rows, all of which have 'hack_type' set to 'Internal'. The rows are numbered 43 through 58.

	hack_id	hack_type
43	43	Internal
44	44	External
45	45	Internal
46	39	Internal
47	35	Internal
48	36	Internal
49	28	Internal
50	29	Internal
51	26	Internal
52	22	Internal
53	23	Internal
54	15	Internal
55	2	Internal
56	3	Internal
57	5	Internal
58	19	Internal

Total rows: 58 of 58 Query complete 00:00:00.077

Number of Tuples → 58

■ Apply (Part_id, Hack_id)

Query Query History

```
68 COPY Apply FROM 'D:\DBMS\final\final\Apply_1.csv' DELIMITER ',' CSV HEADER;
69 select * from Apply
70
71
```

Data output Messages Notifications

A data grid showing the contents of the 'Apply' table. The table has two columns: 'hack_id' [PK] bigint and 'part_id' [PK] bigint. The data consists of 45 rows, each containing a unique 'hack_id' and a corresponding 'part_id'. The rows are numbered from 30 to 45.

	hack_id [PK] bigint	part_id [PK] bigint
30	39	30
31	35	26
32	25	2
33	22	15
34	2	45
35	43	21
36	1	3
37	20	42
38	14	28
39	2	10
40	6	39
41	40	28
42	8	30
43	35	5
44	39	14
45	26	44

Total rows: 45 of 45 Query complete 00:00:00.084 ✓ Successful

Number of Tuples → 45

■ Stipend (Hack_id, Stipend_given, Mode)

Query Query History

```
85 COPY Stipend FROM 'D:\DBMS\final\Stipend_1.csv' DELIMITER ',' CSV HEADER;
86 select * from Stipend
87
88
```

Data output Messages Notifications

	hack_id [PK] bigint	stipend_given [PK] bigint	mode character varying
30	33	37407	Offline
31	32	65629	Offline
32	31	45900	Online
33	29	26552	Offline
34	26	106248	Offline
35	25	87541	Online
36	21	97844	Offline
37	23	15349	Online
38	20	49139	Online
39	13	95880	Offline
40	19	39142	Offline
41	9	34091	Online
42	39	61903	Offline
43	33	87723	Offline
44	23	72840	Online
45	32	48748	Online

Total rows: 45 of 45 Query complete 00:00:00.072

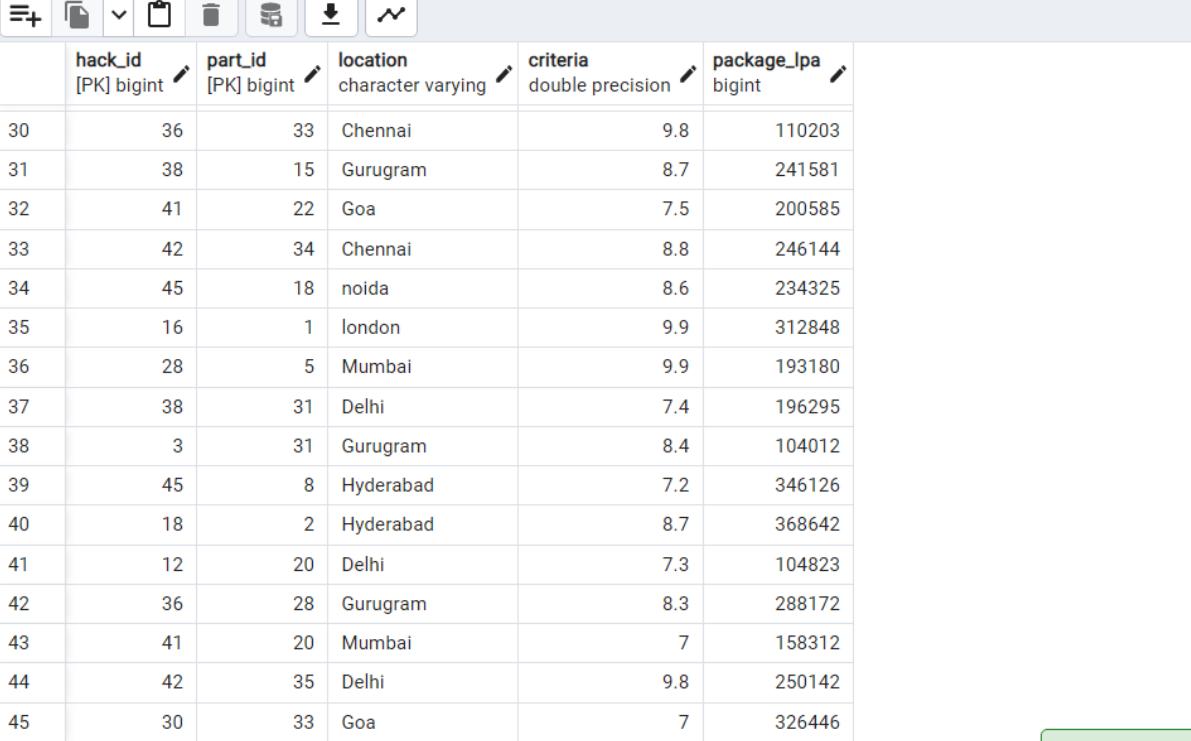
Number of Tuples → 45

■ Package (Hack_id, Part_id, Location, Package_LPA, Criteria)

Query Query History

```
108 COPY Package FROM 'D:\DBMS\final\final\Package_1.csv' DELIMITER ',' CSV HEADER;
109 select * from Package
110
111
```

Data output Messages Notifications



	hack_id [PK] bigint	part_id [PK] bigint	location character varying	criteria double precision	package_lpa bigint
30	36	33	Chennai	9.8	110203
31	38	15	Gurugram	8.7	241581
32	41	22	Goa	7.5	200585
33	42	34	Chennai	8.8	246144
34	45	18	noida	8.6	234325
35	16	1	london	9.9	312848
36	28	5	Mumbai	9.9	193180
37	38	31	Delhi	7.4	196295
38	3	31	Gurugram	8.4	104012
39	45	8	Hyderabad	7.2	346126
40	18	2	Hyderabad	8.7	368642
41	12	20	Delhi	7.3	104823
42	36	28	Gurugram	8.3	288172
43	41	20	Mumbai	7	158312
44	42	35	Delhi	9.8	250142
45	30	33	Goa	7	326446

Total rows: 45 of 45 Query complete 00:00:00.106 ✓ Successful

Number of Tuples → 45

■ Sponsorship (Sponsor_id, Sponsor_Name)

The screenshot shows a database interface with a query editor and a results viewer. The query editor contains the following SQL code:

```
120 COPY Sponsorship FROM 'D:\DBMS\final\final\Sponsorship_1.csv' DELIMITER ',' CSV HEADER;
121 select * from Sponsorship
122
123
```

The results viewer displays the data from the Sponsorship table:

	sponsor_id [PK] bigint	sponsor_name character varying
30	30	Tagcat
31	31	Zoomcast
32	32	Izio
33	33	Centimia
34	34	Ntags
35	35	Eidel
36	36	Ooba
37	37	Dabjam
38	38	Kwinu
39	39	Yabox
40	40	Realpoint
41	41	Yodo
42	42	Abatz
43	43	Thoughtmix
44	44	Skivee
45	45	Flashpoint

Total rows: 45 of 45 Query complete 00:00:00.090 ✓ Successfully run. To

Number of Tuples → 45

■ **Round (Platform, No_of_question, date_of_round, time_of_round, duration_of_round)**

Query History																																																																																																										
135	COPY Round FROM 'D:\DBMS\final\final\Round_1.csv' DELIMITER ',' CSV HEADER;																																																																																																									
136	select * from Round																																																																																																									
137																																																																																																										
138																																																																																																										
Data output Messages Notifications																																																																																																										
<table border="1"> <thead> <tr> <th></th> <th>platform [PK] character varying</th> <th>no_of_question bigint</th> <th>date_of_round date</th> <th>time_of_round time without time zone</th> <th>duration_of_round bigint</th> </tr> </thead> <tbody> <tr><td>30</td><td>Reichert-O'Conner</td><td>4</td><td>2023-01-17</td><td>19:48:00</td><td>4</td></tr> <tr><td>31</td><td>Leetcode</td><td>8</td><td>2023-10-24</td><td>13:16:00</td><td>3</td></tr> <tr><td>32</td><td>Rolfson, Bernier and ...</td><td>10</td><td>2023-04-21</td><td>11:14:00</td><td>3</td></tr> <tr><td>33</td><td>Rolfson-Luettgen</td><td>6</td><td>2023-02-06</td><td>11:23:00</td><td>3</td></tr> <tr><td>34</td><td>Russel and Sons</td><td>5</td><td>2023-01-01</td><td>18:26:00</td><td>2</td></tr> <tr><td>35</td><td>Ryan Group</td><td>10</td><td>2023-01-30</td><td>17:48:00</td><td>4</td></tr> <tr><td>36</td><td>Codeninja</td><td>9</td><td>2023-03-27</td><td>11:43:00</td><td>4</td></tr> <tr><td>37</td><td>Schamberger Group</td><td>6</td><td>2023-03-21</td><td>19:28:00</td><td>4</td></tr> <tr><td>38</td><td>Schuster-Lakin</td><td>5</td><td>2023-02-06</td><td>19:05:00</td><td>2</td></tr> <tr><td>39</td><td>Sipes-Adams</td><td>9</td><td>2023-03-14</td><td>18:36:00</td><td>3</td></tr> <tr><td>40</td><td>Towne Group</td><td>4</td><td>2023-03-16</td><td>14:55:00</td><td>2</td></tr> <tr><td>41</td><td>Codestudio</td><td>7</td><td>2022-11-21</td><td>14:58:00</td><td>2</td></tr> <tr><td>42</td><td>Veum LLC</td><td>3</td><td>2023-08-24</td><td>17:20:00</td><td>3</td></tr> <tr><td>43</td><td>Weimann, Skiles and ...</td><td>4</td><td>2023-06-04</td><td>11:49:00</td><td>2</td></tr> <tr><td>44</td><td>White and Sons</td><td>8</td><td>2023-06-12</td><td>17:10:00</td><td>2</td></tr> <tr><td>45</td><td>Interviewbit</td><td>3</td><td>2023-05-25</td><td>17:51:00</td><td>4</td></tr> </tbody> </table>						platform [PK] character varying	no_of_question bigint	date_of_round date	time_of_round time without time zone	duration_of_round bigint	30	Reichert-O'Conner	4	2023-01-17	19:48:00	4	31	Leetcode	8	2023-10-24	13:16:00	3	32	Rolfson, Bernier and ...	10	2023-04-21	11:14:00	3	33	Rolfson-Luettgen	6	2023-02-06	11:23:00	3	34	Russel and Sons	5	2023-01-01	18:26:00	2	35	Ryan Group	10	2023-01-30	17:48:00	4	36	Codeninja	9	2023-03-27	11:43:00	4	37	Schamberger Group	6	2023-03-21	19:28:00	4	38	Schuster-Lakin	5	2023-02-06	19:05:00	2	39	Sipes-Adams	9	2023-03-14	18:36:00	3	40	Towne Group	4	2023-03-16	14:55:00	2	41	Codestudio	7	2022-11-21	14:58:00	2	42	Veum LLC	3	2023-08-24	17:20:00	3	43	Weimann, Skiles and ...	4	2023-06-04	11:49:00	2	44	White and Sons	8	2023-06-12	17:10:00	2	45	Interviewbit	3	2023-05-25	17:51:00	4
	platform [PK] character varying	no_of_question bigint	date_of_round date	time_of_round time without time zone	duration_of_round bigint																																																																																																					
30	Reichert-O'Conner	4	2023-01-17	19:48:00	4																																																																																																					
31	Leetcode	8	2023-10-24	13:16:00	3																																																																																																					
32	Rolfson, Bernier and ...	10	2023-04-21	11:14:00	3																																																																																																					
33	Rolfson-Luettgen	6	2023-02-06	11:23:00	3																																																																																																					
34	Russel and Sons	5	2023-01-01	18:26:00	2																																																																																																					
35	Ryan Group	10	2023-01-30	17:48:00	4																																																																																																					
36	Codeninja	9	2023-03-27	11:43:00	4																																																																																																					
37	Schamberger Group	6	2023-03-21	19:28:00	4																																																																																																					
38	Schuster-Lakin	5	2023-02-06	19:05:00	2																																																																																																					
39	Sipes-Adams	9	2023-03-14	18:36:00	3																																																																																																					
40	Towne Group	4	2023-03-16	14:55:00	2																																																																																																					
41	Codestudio	7	2022-11-21	14:58:00	2																																																																																																					
42	Veum LLC	3	2023-08-24	17:20:00	3																																																																																																					
43	Weimann, Skiles and ...	4	2023-06-04	11:49:00	2																																																																																																					
44	White and Sons	8	2023-06-12	17:10:00	2																																																																																																					
45	Interviewbit	3	2023-05-25	17:51:00	4																																																																																																					
Total rows: 45 of 45 Query complete 00:00:00.081																																																																																																										
✓ Successfully																																																																																																										

Number of Tuples → 45

■ Mult_Round (Platform, difficulty_of_round)

Query Query History

```
150 COPY Mult_Round FROM 'D:\DBMS\final\final\Mult_Round_1.csv' DELIMITER ',' CSV HEADER;
151 select * from Mult_Round
152
153
```

Data output Messages Notifications

	platform character varying	difficulty_of_round character varying
64	Jacobs-DuBuque	Medium
65	Jacobs-DuBuque	Hard
66	Kub, Quitzon and Windler	Easy
67	Kub, Quitzon and Windler	Hard
68	Codenation	Medium
69	Macejkovic LLC	Easy
70	Marquardt-Nolan	Hard
71	Hackerrank	Easy
72	Monahan, Haley and Smith	Hard
73	Morar LLC	Easy
74	Reichert-O'Conner	Hard
75	Reichert-O'Conner	Easy
76	Leetcode	Hard
77	Sipes-Adams	Easy
78	Interviewbit	Hard
79	Interviewbit	Easy

Total rows: 79 of 79 Query complete 00:00:00.076 ✓ Successfully r

Number of Tuples ➔ 79

■ Kind (Hack_id, Part_id, Platform)

Query Query History

```
174 COPY Kind FROM 'D:\DBMS\final\Kind_1.csv' DELIMITER ',' CSV HEADER;
175 select * from Kind
176
177
```

Data output Messages Notifications

	part_id bigint	lock	hack_id bigint	lock	platform character varying	lock
4	8		10		Conroy, Dibbert and Gerlach	
5	10		12		Crona, Kiehn and Cruicksha...	
6	12		13		Crooks, Heaney and Zieme	
7	14		16		Davis-Baumbach	
8	15		18		Feil Group	
9	17		19		CSES	
10	23		20		Goldner Group	
11	24		21		Codechef	
12	25		23		Heathcote LLC	
13	29		24		Heidenreich-Welch	
14	31		25		Hermann Inc	
15	36		26		Hickle, Kovacek and Fay	
16	37		27		Hickle-Hermiston	
17	43		28		Codeforces	
18	44		29		Jacobs-DuBuque	
19	45		30		Kessler, Bradtke and Mertz	

Total rows: 19 of 19 Query complete 00:00:00.072

Number of Tuples → 19

■ Award (Prize_id, Hack_id, Prize_1, Prize_2, Prize_3)

Query Query History

```
193 COPY Award FROM 'D:\DBMS\final\Award_1.csv' DELIMITER ',' CSV HEADER;
194 select * from Award
195
196
```

Data output Messages Notifications

	prize_id [PK] bigint	hack_id bigint	prize_1 bigint	prize_2 bigint	prize_3 bigint
1	1	1	1000	750	500
2	2	4	4000	3000	2000
3	3	5	5000	3750	2500
4	4	6	6000	4500	3000
5	5	7	7000	5250	3500
6	6	8	8000	6000	4000
7	7	11	11000	8250	5500
8	8	14	14000	10500	7000
9	9	15	15000	11250	7500
10	10	17	17000	12750	8500
11	11	22	22000	16500	11000
12	12	37	37000	27750	18500
13	13	40	40000	30000	20000
14	14	43	43000	32250	21500
15	15	44	44000	33000	22000

Total rows: 15 of 15 Query complete 00:00:00.077 ✓ Success

Number of Tuples → 15

■ Declare (Prize_id, Hack_id, Part_id, First_Name, Last_Name, Winning_Prizes)

Query Query History

```
216 COPY Declare FROM 'D:\DBMS\final\Declare_1.csv' DELIMITER ',' CSV HEADER;
217 select * from Declare
218
219
```

Data output Messages Notifications

	hack_id bigint	prize_id bigint	part_id bigint	first_name character varying	last_name character varying	winning_prizes bigint
1	1	1	44	Darrel	Student	1000
2	4	2	2	Wakefield	Halesworth	4000
3	5	3	5	Monroe	Beneze	5000
4	6	4	8	Roderich	Clissold	6000
5	7	5	10	Joe	Edinboro	7000
6	8	6	12	Stanford	Hampe	8000
7	11	7	14	Guilbert	Basson	11000
8	14	8	43	Franzen	Sallans	14000
9	15	9	17	Jozef	Matis	15000
10	17	10	23	Herschel	Halpeine	17000
11	22	11	24	Gustavus	Prangle	22000
12	37	12	25	Heall	Myers	37000
13	40	13	45	Hermie	Perkins	40000
14	43	14	31	Godfree	Muriel	43000
15	44	15	36	Morry	Goodband	44000

Total rows: 15 of 15 Query complete 00:00:00.083

Number of Tuples ➔ 15

■ Help (Prize_id, Sponsor_id)

Query Query History

```
235 COPY Help FROM 'D:\DBMS\final\Help_1.csv' DELIMITER ',' CSV HEADER;
236 select * from Help
237
238 ----- 1
```

Data output Messages Notifications



	prize_id	sponsor_id
1	1	45
2	2	7
3	3	16
4	4	18
5	5	10
6	6	4
7	7	22
8	8	27
9	9	25
10	10	37
11	11	33
12	12	3
13	13	6
14	14	9
15	15	12

Total rows: 15 of 15 Query complete 00:00:00.051

Number of Tuples ➔ 15

SQL Queries:

Query-1.

Use of between.

SQL Query-1.

```
select Part_id, First_Name, Last_Name, DOB, CPI
from participants
where CPI between 6.5 and 9.3;
```

Snapshot-1.

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query' and 'Query History'. Below the tabs, the query code is displayed:

```
238 -----1-----
239
240 select Part_id, First_Name, Last_Name, DOB, CPI
241 from participants
242 where CPI between 6.5 and 9.3;
243
```

Below the code, there are three tabs: 'Data output', 'Messages', and 'Notifications'. The 'Data output' tab is selected and shows a table with the query results. The table has columns: part_id [PK] bigint, first_name character varying, last_name character varying, dob date, and cpi double precision. The data consists of 24 rows, each representing a participant. The last row of the table is highlighted in light blue. At the bottom of the results table, it says 'Total rows: 24 of 24' and 'Query complete 00:00:00.077'.

	part_id [PK] bigint	first_name character varying	last_name character varying	dob date	cpi double precision
11	22	Sabra	Symms	1998-06-08	7.62
12	26	Marlee	Voas	2000-12-16	8.57
13	29	Francis	Gery	1988-01-01	8.08
14	33	Denna	Kelwick	1999-02-21	8.94
15	34	Eolande	Earpe	1993-07-17	7.04
16	37	Buck	Chappelow	1990-11-21	8.47
17	38	Mirabel	Bunton	1992-01-03	7.44
18	39	Rosalinda	Dinjes	1991-04-19	7.47
19	40	Shoshana	Gallelli	2000-07-08	6.76
20	41	Lurline	Brackenbury	2003-02-28	8.88
21	42	Vevay	Itzkovwitch	1990-02-22	9.17
22	43	Franzen	Sallans	1991-09-23	6.84
23	44	Darrel	Student	1995-12-12	7.35
24	45	Hermie	Perkins	2004-01-05	6.56

Number of Tuples → 24

Query-2.

Use of String % and Condition.

SQL Query-2.

```
select Hack_id, Part_id, First_Name, Last_Name, Winning_Prizes  
from declare  
where First_Name like 'H%' or Last_Name like 'H%' and  
Winning_Prizes >= 15000;
```

Snapshot-2.

The screenshot shows a SQL query editor interface. The top navigation bar includes 'Query' (selected), 'Query History', and other tabs. Below the tabs, the query text is displayed:

```
244 -----2-----  
245  
246 select Hack_id, Part_id, First_Name, Last_Name, Winning_Prizes  
247 from declare  
248 where First_Name like 'H%' or Last_Name like 'H%' and Winning_Prizes >= 15000;  
249
```

Below the query text, there are tabs for 'Data output', 'Messages', and 'Notifications'. The 'Data output' tab is selected, showing a table with the following data:

	hack_id	part_id	first_name	last_name	winning_prizes
1	17	23	Herschel	Halpeine	17000
2	37	25	Heall	Myers	37000
3	40	45	Hermie	Perkins	40000

At the bottom of the interface, status information is shown: 'Total rows: 3 of 3', 'Query complete 00:00:00.089', and a green success message 'Success!'. There is also a small icon of a checkmark.

Number of Tuples → 3

Query-3.

Use of Group by, Having, Order by and count function.

SQL Query-3.

```
select Name_Type, count(*) as Count  
from Hackathon  
group by Name_Type  
Having count(*) >= 14  
order by Name_Type DESC;
```

Snapshot-3.

The screenshot shows a SQL query editor interface. The top navigation bar includes 'Query' (selected), 'Query History', and other tabs. Below the tabs, the query code is displayed:

```
249  
250 -----3-----  
251  
252 select Name_Type, count(*) as Count  
253 from Hackathon  
254 group by Name_Type  
255 Having count(*) >= 14  
256 order by Name_Type DESC;  
257
```

The code is highlighted in blue and purple. The result set is shown below the code:

	name_type	count
1	Programming	15
2	Job	17

At the bottom, status information is provided: 'Total rows: 2 of 2' and 'Query complete 00:00:00.076'. A green checkmark icon with the word 'Success' is visible on the right.

Number of Tuples → 2

Query-4.

Use of between and order by. Compare foreign key.

SQL Query-4.

```
select H.Hack_Name, H.Date, H.Time, H.Duration, P.Location, P.Criteria,
P.Package_LPA
from Hackathon as H, Package as P
where H.Hack_id = P.Hack_id and Criteria between 7.5 and 8.9
order by P.Location;
```

Snapshot-4.

The screenshot shows a SQL query editor interface. The top section displays the query code:

```
257
258 -----4-----
259
260 select H.Hack_Name, H.Date, H.Time, H.Duration, P.Location, P.Criteria, P.Package_LPA
261 from Hackathon as H, Package as P
262 where H.Hack_id = P.Hack_id and Criteria between 7.5 and 8.9
263 order by P.Location;
264
265 -----5-----
```

The bottom section shows the resulting data output:

	hack_name	date	time	duration	location	criteria	package_lpa
13	Zoolab	2023-07-31	10:05:00	2	Hyderabad	8.4	334211
14	Sonair	2023-03-26	10:06:00	4	Hyderabad	8.7	368642
15	Namfix	2023-06-18	12:55:00	2	london	8.4	284029
16	Home Ing	2023-10-22	16:20:00	3	london	8.1	242937
17	Sonair	2023-03-26	10:06:00	4	london	7.6	383938
18	Zoolab	2023-07-31	16:05:00	2	Mumbai	8.8	101309
19	Toughjoyfax	2023-08-12	13:00:00	2	Mumbai	7.5	183576
20	Zontrax	2023-09-03	12:16:00	4	Mumbai	7.9	202368
21	Wrapsafe	2023-11-11	13:59:00	3	noida	8.6	234325
22	Konklux	2023-06-09	12:56:00	2	noida	8.3	118703
23	Keylex	2022-12-09	09:19:00	3	noida	8.4	224846
24	Keylex	2022-12-09	09:19:00	3	Pune	7.7	351650
25	Stronghold	2023-06-29	11:32:00	2	Pune	8.7	331260

Total rows: 25 of 25 Query complete 00:00:00.074

Number of Tuples → 25

Query-5.

Use of Natural Join with order by and limit function.

SQL Query-5.

```
select MR.Platform, MR.difficulty_of_round, R.No_of_question,
R.Date_of_round, R.Time_of_round
from Round as R Natural Join Mult_Round as MR
where R.No_of_question between 4 and 8
order by R.Date_of_round
limit 20;
```

Snapshot-5.

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query' and 'Query History'. Below the tabs, the query number '265' is followed by a dashed orange line, and the query itself is displayed:

```
265 -----5-----
266
267 select MR.Platform, MR.difficulty_of_round, R.No_of_question, R.Date_of_round, R.Time_of_round
268 from Round as R Natural Join Mult_Round as MR
269 where R.No_of_question between 4 and 8
270 order by R.Date_of_round
271 limit 20;
272
```

Below the query, there are buttons for 'Data output', 'Messages', and 'Notifications'. The main area displays the results of the query as a table:

	platform	difficulty_of_round	no_of_question	date_of_round	time_of_round
8	Reichert-O'Connor	Easy	4	2023-01-17	19:48:00
9	Monahan, Haley ...	Hard	5	2023-01-21	10:36:00
10	Monahan, Haley ...	Medium	5	2023-01-21	10:36:00
11	Schuster-Lakin	Hard	5	2023-02-06	19:05:00
12	Rolfson-Luettgen	Easy	6	2023-02-06	11:23:00
13	Jacobs-DuBuque	Medium	4	2023-02-12	14:03:00
14	Jacobs-DuBuque	Easy	4	2023-02-12	14:03:00
15	Jacobs-DuBuque	Hard	4	2023-02-12	14:03:00
16	Hackerrank	Medium	7	2023-02-15	18:21:00
17	Hackerrank	Easy	7	2023-02-15	18:21:00
18	Towne Group	Medium	4	2023-03-16	14:55:00
19	Schamberger Gro...	Easy	6	2023-03-21	19:28:00
20	Crooks, Heaney a...	Hard	8	2023-03-22	16:13:00

At the bottom of the results table, it says 'Total rows: 20 of 20'. Below the table, a status bar indicates 'Query complete 00:00:00.088'.

Number of Tuples → 20

Query-6.

Use of 3 Natural Join between Four Table.

SQL Query-6.

```
select H.Hack_Name, He.sponsor_id, S.sponsor_name, A.Prize_1,
A.Prize_2, A.Prize_3
from Award as A Natural Join Hackathon as H Natural Join Help as He
Natural Join Sponsorship as S
```

Snapshot-6.

The screenshot shows a SQL query editor with the following details:

- Query History:** Shows previous queries numbered 272 through 277. Query 275 is highlighted in blue, containing the executed SQL code.
- Code:**

```
272
273 -----6-----
274
275 select H.Hack_Name, He.sponsor_id, S.sponsor_name, A.Prize_1,
276   from Award as A Natural Join Hackathon as H Natural Join Help as He
277 Natural Join Sponsorship as S
```
- Data Output:** The results of the query are displayed in a table with 15 rows. The columns are:

	hack_name	sponsor_id	sponsor_name	prize_1	prize_2	prize_3
1	Fixtirex	45	Flaspoint	1000	750	500
2	Fintone	7	Jetwire	4000	3000	2000
3	Y-find	16	Browsertype	5000	3750	2500
4	Alphazap	18	Talane	6000	4500	3000
5	Cardify	10	Skiptube	7000	5250	3500
6	Greenlam	4	Buzzster	8000	6000	4000
7	Bitwolf	22	Mydo	11000	8250	5500
8	Stringtough	27	Teklist	14000	10500	7000
9	Solarbreeze	25	Kazu	15000	11250	7500
10	Bytecard	37	Dabjam	17000	12750	8500
11	Asoka	33	Centimia	22000	16500	11000
12	It	3	Talane	37000	27750	18500
13	Mat Lam Tam	6	Photolist	40000	30000	20000
14	Rank	9	Zoomdog	43000	32250	21500
15	Opela	12	Dabtype	44000	33000	22000
- Messages:** Shows "Total rows: 15 of 15" and "Query complete 00:00:00.090".
- Status:** Shows a green checkmark icon and the message "Successfully run. Total query runtime".

Number of Tuples → 15

Query-7.

Use of Min, Max, Avg aggregate function with Natural Join, group by and order by clause.

SQL Query-7.

```
select S.Hack_id, H.Hack_Name, avg(S.stipend_given) as
Avarege_Stipend, max(S.stipend_given) as Max_Stipend,
min(S.stipend_given) as Min_Stipend
from Hackathon as H Natural Join Stipend as S
group by S.Hack_id, H.Hack_Name
order by Avarege_Stipend DESC;
```

Snapshot-7.

The screenshot shows a SQL query editor interface. The top pane displays the SQL code:

```
278
279
280 select S.Hack_id, H.Hack_Name, avg(S.stipend_given) as Avarege_Stipend, max(S.stipend_given) as Max_Stipend, min(S.stipend_given)
281 from Hackathon as H Natural Join Stipend as S
282 group by S.Hack_id, H.Hack_Name
283 order by Avarege_Stipend DESC;
```

The bottom pane shows the resulting data output:

	hack_id	hack_name	avarege_stipend	max_stipend	min_stipend
1	13	Bitchip	100623.000000000	105366	95880
2	25	Kanlam	89578.666666666	106870	74325
3	19	Bigtax	80811.000000000	106416	39142
4	32	Alpha	71218.600000000	87046	48748
5	26	Stringtough	69272.500000000	109235	28799
6	23	Wrapsafe	63060.000000000	91562	15349
7	39	Alphazap	56635.000000000	92176	26575
8	21	Konklab	54932.000000000	97844	19791
9	33	Pannier	53719.500000000	87723	15821
10	20	Sonsing	50908.000000000	67811	35774
11	31	It	49998.000000000	77824	26270
12	29	Toughjoyfax	43922.000000000	61292	26552
13	9	Otcom	30076.500000000	36559	15167

Total rows: 13 of 13 Query complete 00:00:00.062 Ln 280, Col 1

Number of Tuples → 13

Query-8.

Use of Min, Max, Avg aggregate function with Natural Join, group by and order by clause.

SQL Query-8.

```
select Pa.Location, avg(Pa.Package_LPA) as Avarage_Package,
min(Pa.Package_LPA) as Min_Package, max(Pa.Package_LPA) as
Max_Package
from Package as Pa Natural Join Participants as P Natural Join Hackathon
as H
group by Pa.Location
order by Pa.Location;
```

Snapshot-8.

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query' (which is selected) and 'Query History'. Below the tabs, the query code is displayed:

```
285
286
287 select Pa.Location, avg(Pa.Package_LPA) as Avarage_Package, min(Pa.Package_LPA) as Min_Package, max(Pa.Package_LPA) as Max_Package
288 from Package as Pa Natural Join Participants as P Natural Join Hackathon as H
289 group by Pa.Location
290 order by Pa.Location;
291
```

Below the code, there are three tabs: 'Data output', 'Messages', and 'Notifications'. The 'Data output' tab is selected and displays a table with the query results:

	location character varying	avarage_package numeric	min_package bigint	max_package bigint
1	Bengaluru	230773.000000000000	137558	323988
2	Chennai	201977.250000000000	110203	274796
3	Delhi	245719.3333333333	104823	371306
4	Goa	258962.000000000000	198452	354876
5	Gurugram	199909.000000000000	104012	288172
6	Hyderabad	338649.000000000000	306362	368642
7	london	305938.000000000000	242937	383938
8	Mumbai	167749.000000000000	101309	202368
9	noida	224310.000000000000	118703	319366
10	Pune	241420.750000000000	113057	351650

At the bottom of the results pane, it says 'Total rows: 10 of 10' and 'Query complete 00:00:00.089'. To the right, it says 'Ln 287, Col 1'.

Number of Tuples → 10

Query-9.

Use of Left Join.

SQL Query-9.

*SELECT **

from Participants as P Left Join Kind as K

ON K.Part_id = P.Part_id;

Snapshot-9.

part_id	first_name	last_name	email_add	dob	student	college	cpi	part_id	hack_id	platform
33	38	Mirabel	Bunton	mbunton11@sta...	1992-01-03	Employee	Boston University	7.44	[null]	[null]
34	28	Adelaide	Troder	atrodler@unesco...	1991-06-16	Employee	University of Nort...	4.06	[null]	[null]
35	30	Antoinette	Perigeaux	aperigeaux@yell...	1997-03-16	Employee	Universidad Agric...	9.48	[null]	[null]
36	42	Vevay	Itzkovwitch	vitzkovwitch15@...	1990-02-22	Employee	University of Dayton	9.17	[null]	[null]
37	6	Idette	Harback	iharback5@seattl...	2000-10-15	Employee	Shanghai Institute of...	7.78	[null]	[null]
38	41	Lurline	Brackenbury	lbrackenbury14@...	2003-02-28	Employee	Rochester Institute of...	8.88	[null]	[null]
39	16	Juditha	Watman	jwatmanf@earthlink...	1995-04-13	Employee	University of Ottawa	9.82	[null]	[null]
40	4	Silvia	Calvey	scalvey3@pbs.org	2002-12-30	Employee	Georg-Simon-Ohm...	9.39	[null]	[null]
41	22	Sabra	Symms	ssymms1@360.cn	1998-06-08	Employee	Lenoir-Rhyne College	7.62	[null]	[null]
42	3	Jerrie	Widdowson	jwiddowson2@p...	1995-04-23	Employee	Osaka University ...	9.85	[null]	[null]
43	35	Karleen	Beinke	kbeinkey@phpb...com	1989-06-01	Employee	Kyungpook National University	4.23	[null]	[null]
44	9	Pearl	Hornung	phornung8@umich...	1987-01-04	Employee	Dankook University	6.51	[null]	[null]
45	7	Sabine	Butterley	sbutterley6@earth...	1998-12-24	Employee	University of North Carolina at Chapel Hill	9.15	[null]	[null]

Number of Tuples → 45

Query-10.

Use of Right Join.

SQL Query-10.

Select *

from Award as A ***Right Join*** Hackathon as H
on H.Hack_id = A.Hack_id;

Snapshot-10.

```

29 / 298 ----- 10 -----
299
300 Select *
301 from Award as A right Join Hackathon as H
302 on H.Hack_id = A.Hack_id;
303
Data output Messages Notifications

```

prize_id	hack_id	prize_1	prize_2	prize_3	hack_id	hack_name	name_type	date	time	duration
29	[null]	[null]	[null]	[null]	24	Zooiau	Jou	2023-07-31	10:05:00	2
33	[null]	[null]	[null]	[null]	38	Tempsoft	Job	2023-01-28	15:52:00	4
34	[null]	[null]	[null]	[null]	28	Vagram	Job	2023-08-27	14:13:00	3
35	[null]	[null]	[null]	[null]	30	Home Ing	Job	2023-10-22	16:20:00	3
36	[null]	[null]	[null]	[null]	42	Otcom	Job	2023-03-06	12:44:00	3
37	[null]	[null]	[null]	[null]	29	Toughjoyfax	Internship	2022-12-25	11:17:00	4
38	[null]	[null]	[null]	[null]	41	Cardguard	Job	2023-06-20	13:04:00	4
39	[null]	[null]	[null]	[null]	16	Keylex	Job	2022-12-09	09:19:00	3
40	[null]	[null]	[null]	[null]	36	Redhold	Job	2023-11-01	14:58:00	3
41	[null]	[null]	[null]	[null]	23	Wrapsafe	Internship	2023-01-05	14:36:00	2
42	[null]	[null]	[null]	[null]	45	Wrapsafe	Job	2023-11-11	13:59:00	3
43	[null]	[null]	[null]	[null]	3	Namfix	Job	2023-06-18	12:55:00	2
44	[null]	[null]	[null]	[null]	35	Gembucket	Job	2023-04-03	18:43:00	3
45	[null]	[null]	[null]	[null]	9	Otcom	Internship	2023-06-16	14:03:00	4

Total rows: 45 of 45 Query complete 00:00:00.073 ✓ Successfully run. Total query runtime: 73 msec. 45 rows affected. 1

Number of Tuples → 45

Query-11.

Use of Nested Query with Aggregate Function Max and or condition.

SQL Query-11.

```
select *
from Participants
where CPI = (select Max(CPI) as max_CPI
              from Participants) or
          CPI = (select min(CPI) as min_CPI
              from Participants)
```

order by dob;

Snapshot-11.

The screenshot shows a database query editor interface. The top section displays the SQL code:

```
304 -----11-----
305
306 select *
307 from Participants
308 where CPI = (select Max(CPI) as max_CPI
309           from Participants) or
310       CPI = (select min(CPI) as min_CPI
311           from Participants)
312 order by dob;
313
```

The bottom section shows the results of the query:

	part_id	first_name	last_name	email_add	dob	student	college	cpi
1	28	Adelaide	Troder	atrodterr@unes...	1991-06-16	Employee	University of Nort...	4.06
2	3	Jerrie	Widdowson	jwiddowson2@p...	1995-04-23	Employee	Osaka University ...	9.85

Total rows: 2 of 2 Query complete 00:00:00.092 ✓ Successfully run. Total query runtime: 92 ms

Number of Tuples → 2

Query-12.

Use of Nested Query with Aggregate Function Max and or condition.

SQL Query-12.

```
select *
from Hackathon as H Natural Join Award as A
where Prize_1 = (select max(Prize_1) from Award) or
      Prize_2 = (select min(Prize_2) from Award) or
      Prize_3 = (select max(Prize_3) from Award)
order by date;
```

Snapshot-12.

The screenshot shows a SQL query editor interface. The top section displays the query code:

```
314 -----12-----
315
316 select *
317 from Hackathon as H Natural Join Award as A
318 where Prize_1 = (select max(Prize_1) from Award) or
319      Prize_2 = (select min(Prize_2) from Award) or
320      Prize_3 = (select max(Prize_3) from Award)
321 order by date;
322
323 -----13-----
```

The bottom section shows the results of the query execution:

	hack_id	lock	hack_name	lock	name_type	lock	date	lock	time	lock	duration	lock	prize_id	lock	prize_1	lock	prize_2	lock	prize_3	lock
1			Fixflex		Programming		2023-10-03		19:19:00		4		1		1000		750		500	
2			Opela		Programming		2023-10-05		17:17:00		2		15		44000		33000		22000	

Total rows: 2 of 2 Query complete 00:00:00.073 ✓ Successfully run. Total query runtime: 73 msec. 2 rows

Number of Tuples → 2

Query-13.

Use of Natural Join and Sort by Two Attribute of Table.

SQL Query-13.

```
select P.Location, count(*) as Count, max(P.Package_LPA) as
Max_Package, Min(P.Package_LPA) as Min_Package,
Avg(P.Package_LPA) as Avg_Package
from Hackathon as H Natural Join Package as P
group by P.Location
order by Count, Avg_Package;
```

Snapshot-13.

The screenshot shows a SQL query being run in a database environment. The query is as follows:

```
322
323 -----13-----
324
325 select P.Location, count(*) as Count, max(P.Package_LPA) as Max_Package, Min(P.Package_LPA) as Min_Package, Avg(P.Package_LPA) as Avg_Package
326 from Hackathon as H Natural Join Package as P
327 group by P.Location
328 order by Count, Avg_Package;
329
```

The results grid displays the following data:

	location	count	max_package	min_package	avg_package
1	Bengaluru	2	323988	137558	230773.000000
2	Chennai	4	274796	110203	201977.250000
3	noida	4	319366	118703	224310.000000
4	Pune	4	351650	113057	241420.750000
5	london	4	383938	242937	305938.000000
6	Mumbai	5	202368	101309	167749.000000
7	Goa	5	354876	198452	258962.000000
8	Hyderabad	5	368642	306362	338649.000000
9	Gurugram	6	288172	104012	199909.000000
10	Delhi	6	371306	104823	245719.333333

Total rows: 10 of 10 Query complete 00:00:00.071 ✓ Successfully run. Total query runtime: 71 msec. 10 rows affected.

Number of Tuples → 10

Query-14.

Use of IN and Two Natural Join between Three Table.

SQL Query-14.

```
select *
```

```
from Participants as P Natural Join Apply as A Natural Join Hackathon as H
```

```
where Student IN (select Student
```

```
from Participants
```

```
where Student = 'Employee')
```

```
order by CPI DESC;
```

Snapshot-14.

```
Query History
330 -----14-----
331
332 select *
333 from Participants as P Natural Join Apply as A Natural Join Hackathon as H
334 where Student IN (select Student
335     from Participants
336     where Student = 'Employee')
337 order by CPI DESC;
338
```

	hack_id	part_id	first_name	last_name	email_add	dob	student	college	cpi	hack_name
15	12	39	Rosalinda	Dinjes	rdinjes12@ehow...	1991-04-19	Employee	Universidad de Z...	7.47	Konklux
16	27	38	Mirabel	Bunton	mbunton11@sta...	1992-01-03	Employee	Boston University	7.44	Alphazap
17	44	40	Shoshana	Gallelli	sgallelli13@icq.c...	2000-07-08	Employee	Universidad Tecn...	6.76	Opela
18	10	27	Joanie	Epsley	jepsleyq@disqus...	1992-01-10	Employee	California State U...	5.52	Toughjoyfax
19	22	32	Katey	Ethridge	kethridgev@artis...	2002-05-21	Employee	Bethel College St...	5.09	Asoka
20	36	13	Coriss	Antonopoulos	cantonopoulosc...	2003-08-06	Employee	Rowan University	5.08	Redhold
21	43	21	Charmain	Brewis	cbrewisk@360.cn	1990-02-03	Employee	American Univer...	4.56	Rank
22	36	21	Charmain	Brewis	cbrewisk@360.cn	1990-02-03	Employee	American Univer...	4.56	Redhold
23	4	35	Karleen	Beinke	kbeinkey@phpbb...	1989-06-01	Employee	Kyungpook Natio...	4.23	Fintone
24	14	28	Adelaide	Trodler	atrodler@unesc...	1991-06-16	Employee	University of Nort...	4.06	Stringtough
25	40	28	Adelaide	Trodler	atrodler@unesc...	1991-06-16	Employee	University of Nort...	4.06	Mat Lam Tam

Total rows: 25 of 25 Query complete 00:00:00.072 Ln 332, Col 1

Number of Tuples → 25

Query-15.

Use of Count Aggregate function and group by, order by clause.

SQL Query-15.

```
select Hack_Type, count(*)
```

```
from Mult_Hackathon
```

```
group by Hack_Type
```

```
order by Hack_Type;
```

Snapshot-15.

The screenshot shows a SQL query editor interface. The top menu bar has 'Query' and 'Query History'. The main area displays a numbered list of SQL statements (339 to 346) and their execution status. Statements 341 through 345 are highlighted in blue, indicating they have been run. Statement 346 is currently being run, as indicated by the progress bar below it. Below the statements is a table titled 'Data output' showing the results of the query. The table has two columns: 'hack_type' (character varying) and 'count' (bigint). It contains two rows: one for 'External' with a count of 26, and one for 'Internal' with a count of 32. At the bottom of the editor, there is a message bar stating 'Total rows: 2 of 2' and 'Query complete 00:00:00.087'. A green 'Successful' status indicator is visible on the right.

	hack_type	count
1	External	26
2	Internal	32

Number of Tuples → 2

Query-16.

Use of View and Nested Queries in View.

SQL Query-16.

create or replace view Prize_Sum as

```
select H.Hack_Name, D.First_Name, D.Last_Name, P.Email_add,
D.Winning_Prizes
```

*from declare as D Natural Join Participants as P Natural Join Hackathon
as H*

where D.Winning_Prizes > (select avg(Winning_Prizes) from declare);

Snapshot-16.

The screenshot shows a SQL query editor interface. At the top, there are tabs for 'Query' and 'Query History'. Below the tabs, the code for creating a view is displayed:

```
345
346 -----16-----
347
348 create or replace view Prize_Sum as
349 select H.Hack_Name, D.First_Name, D.Last_Name, P.Email_add, D.Winning_Prizes
350 from declare as D Natural Join Participants as P Natural Join Hackathon as H
351 where D.Winning_Prizes > (select avg(Winning_Prizes) from declare);
352
```

Below the code, there are buttons for 'Data output', 'Messages', and 'Notifications'. The 'Data output' tab is selected, showing a table with the following data:

	hack_name character varying	first_name character varying	last_name character varying	email_add character varying	winning_prizes bigint
1	Asoka	Gustavus	Prangle	gpranglen@blogs.com	22000
2	It	Heall	Myers	hmyerso@multiply.com	37000
3	Mat Lam Tam	Hermie	Perkins	hperkins18@state.gov	40000
4	Rank	Godfree	Muriel	gmurielu@artisteer.co...	43000
5	Opela	Morry	Goodband	mgoodbandz@xrea.c...	44000

At the bottom of the results pane, it says 'Total rows: 5 of 5' and 'Query complete 00:00:00.070'.

Number of Tuples → 5

Query-17.

Use Prize_Sum View and Run the Query.

SQL Query-17.

```
select *  
from Prize_Sum  
order by Winning_Prizes desc  
limit 3;
```

Snapshot-17.

The screenshot shows a MySQL Workbench interface. The 'Query' tab is selected, displaying the following SQL code:

```
352  
353 -----17-----  
354  
355 select *  
356 from Prize_Sum  
357 order by Winning_Prizes desc  
358 limit 3;  
359
```

Below the code, the results are displayed in a table:

	hack_name character varying	first_name character varying	last_name character varying	email_add character varying	winning_prizes bigint
1	Opela	Morry	Goodband	mgoodbandz@xrea.c...	44000
2	Rank	Godfree	Muriel	gmurielu@artisteer.co...	43000
3	Mat Lam Tam	Hermie	Perkins	hperkins18@state.gov	40000

At the bottom of the interface, status information is shown: 'Total rows: 3 of 3' and 'Query complete 00:00:00.069'. To the right, a green button indicates success with a checkmark icon.

Number of Tuples → 3

Query-18.

Create a function that returns the table containing the Hack_id, Prize_id, Hack_Name, First_Name, Last_Name, Winning_Prizes by Natural Join of Two Table.

SQL Query-18.

```
create or replace function Hackathon_System()
returns table(Hack_id bigint, Prize_id bigint, Hack_Name varchar,
First_Name varchar, Last_Name varchar, Winning_Prizes bigint)
language 'plpgsql'
as $body$
begin
return query execute format('select Hack_id, Prize_id, Hack_Name,
First_Name, Last_Name, Winning_Prizes from Hackathon natural join
declare order by Winning_Prizes DESC');
end
$body$
```



```
select *
from Hackathon_System();
```

Snapshot-18.

Query Query History

```
360 -----18-----
361
362 create or replace function Hackathon_System()
363 returns table(Hack_id bigint, Prize_id bigint, Hack_Name varchar, First_Name varchar, Last_Name varchar, Winning_Prizes bigint)
364 language 'plpgsql'
365 as $body$
366 begin
367 return query execute format('select Hack_id, Prize_id, Hack_Name, First_Name, Last_Name, Winning_Prizes from Hackathon
368                               natural join declare order by Winning_Prizes DESC');
369 end
$body$
```

370
371
372 select *
373 from Hackathon_System();
374

Data output Messages Notifications

	hack_id bigint	prize_id bigint	hack_name character varying	first_name character varying	last_name character varying	winning_prizes bigint
1	44	15	Opela	Morry	Goodband	44000
2	43	14	Rank	Godfree	Muriel	43000
3	40	13	Mat Lam Tarn	Hermie	Perkins	40000
4	37	12	It	Heall	Myers	37000
5	22	11	Asoka	Gustavus	Prangle	22000
6	17	10	Bytecard	Herschel	Halpeine	17000
7	15	9	Solarbreeze	Jozef	Matis	15000

Total rows: 15 of 15 Query complete 00:00:00.067 Ln 372, Col 1

Number of Tuples → 15

Query-19.

Create a trigger that checks the primary key constraints before inserting a new record in the Participants table.

SQL Query-19.

```
create or replace function func_1()
returns trigger
language 'plpgsql'
as $body$
begin
if new.Part_id in (select Part_id from Participants) then
raise notice 'Violating primary key constraints: Part_id = % already
exists', new.Part_id;
return old;
else
raise notice 'Record inserted successfully!';
return new;
end if;
end
$body$
```

```
create or replace trigger trigger_1
before insert
on Participants
for each row
execute procedure func_1();
select * from Participants;
```

Snapshot-19.

Data output Messages Notifications

	part_id [PK] bigint	first_name character varying	last_name character varying	email_addr character varying	dob date	student character varying	college character varying	cpi double precision
25	25	Heall	Myers	hmymerso@multip...	1989-01-08	Student	Galilee College	5.96
26	26	Marlee	Voas	mvoasp@cornell....	2000-12-16	Employee	Allahabad Univer...	8.57
27	27	Joanie	Epsley	jepsleyq@disqus....	1992-01-10	Employee	California State U...	5.52
28	28	Adelaide	Trodlar	atrodler@unesco...	1991-06-16	Employee	University of Nort...	4.06
29	29	Francis	Gery	fgerys@imdb.com	1988-01-01	Student	Universidad Tecn...	8.08
30	30	Antoinette	Perigeaux	aperigeauxt@yel...	1997-03-16	Employee	Universidad Agra...	9.48
31	31	Godfree	Muriel	gmurielu@artiste...	1992-07-26	Student	Universidad Pont...	5.54
32	32	Katey	Ethridge	kethrlidge@artis...	2002-05-21	Employee	Bethel College St...	5.09
33	33	Denna	Kelwick	dkelwickw@unes...	1999-02-21	Employee	American Univer...	8.94
34	34	Eolande	Earp	earpex@wooth...e...	1993-07-17	Employee	Northeast Norma...	7.04
35	35	Karleen	Beinke	kbeinkey@phpbb...	1989-06-01	Employee	Kyungpook Natio...	4.23
36	36	Morry	Goodband	mgoodbandz@xr...	1990-10-13	Student	Sarhad University...	9.42
37	37	Buck	Chappelow	bchappelow10@...	1990-11-21	Student	Mohammad Ali J...	8.47
38	38	Mirabel	Bunton	mbunton11@sta...	1992-01-03	Employee	Boston University	7.44
39	39	Rosalinda	Dinjes	rdinjes12@ehow...	1991-04-19	Employee	Universidad de Z...	7.47
40	40	Shoshana	Gallelli	sgallelli13@icq.c...	2000-07-08	Employee	Universidad Tecn...	6.76
41	41	Lurline	Brackenbury	lbrackenbury14@...	2003-02-28	Employee	Rochester Institu...	8.88
42	42	Vevay	Itzkovwitch	vitzkovwitch15@...	1990-02-22	Employee	University of Dayt...	9.17
43	43	Franzen	Sallans	fsallans16@vina...	1991-09-23	Student	Sacred Heart Uni...	6.84
44	44	Darrel	Student	dStudent17@tiny...	1995-12-12	Student	Northwood Unive...	7.35
45	45	Hermie	Perkins	hperkins18@stat...	2004-01-05	Student	Northern Arizona...	6.56

Total rows: 45 of 45 Query complete 00:00:30.323

Insert into Participants

```
values(44, 'Sahil', 'Mangukiya', 'sahilmangukiya@gmial.com', '2002-09-20', 'Student', 'DAIICT', 7.17) ← Already Exists in Part_id
```

Query Query History

```
400
401  insert into Participants
402  values(44, 'Sahil', 'Mangukiya', 'sahilmangukiya@gmial.com', '2002-09-20', 'Student', 'DAIICT', 7.17)
403
404
```

Data output Messages Notifications

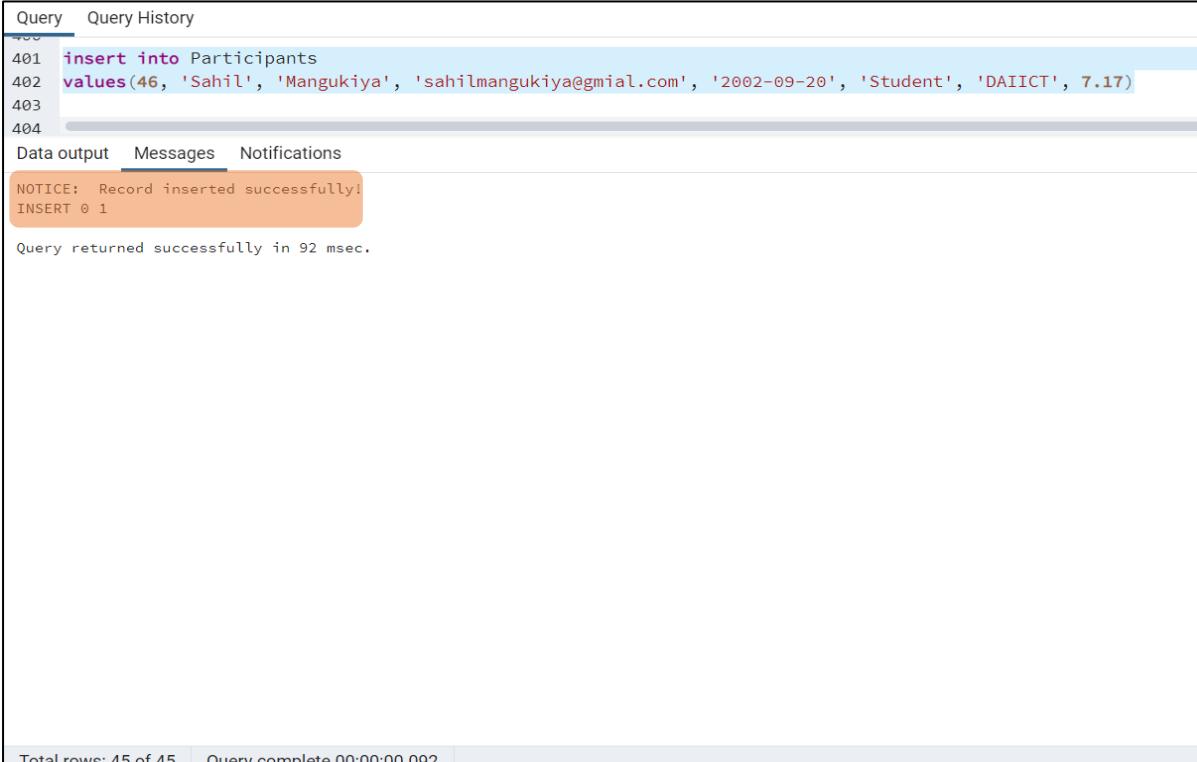
NOTICE: Violating primary key constraints: Part_id = 44 already exists
INSERT 0 0

Query returned successfully in 98 msec.

Total rows: 45 of 45 Query complete 00:00:00.098

Number of Tuples → 45

*insert into Participants
values(46, 'Sahil', 'Mangukiya', 'sahilmangukiya@gmial.com', '2002-09-20', 'Student', 'DAIICT', 7.17) ← Not Exists in Part_id*



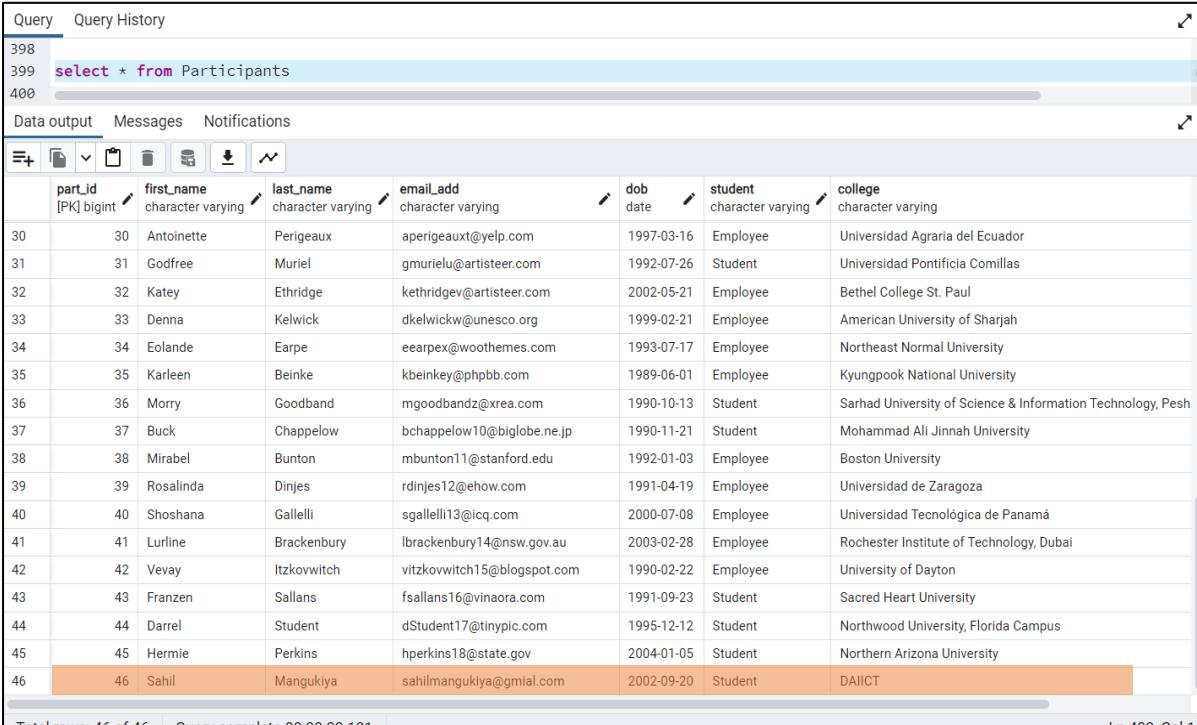
```

Query   Query History
401 insert into Participants
402 values(46, 'Sahil', 'Mangukiya', 'sahilmangukiya@gmial.com', '2002-09-20', 'Student', 'DAIICT', 7.17)
403
404
Data output  Messages  Notifications
NOTICE: Record inserted successfully!
INSERT 0 1
Query returned successfully in 92 msec.

Total rows: 45 of 45  Query complete 00:00:00.092

```

*select * from Participants;*



```

Query   Query History
398
399 select * from Participants
400
Data output  Messages  Notifications
part_id [PK] bigint ✓ first_name character varying ✓ last_name character varying ✓ email_add character varying ✓ dob date ✓ student character varying ✓ college character varying
30 30 Antoinette Perigeaux aperigeaux@yelp.com 1997-03-16 Employee Universidad Agraria del Ecuador
31 31 Godfree Muriel gmurielu@artsteer.com 1992-07-26 Student Universidad Pontificia Comillas
32 32 Kately Ethridge kethridgev@artsteer.com 2002-05-21 Employee Bethel College St. Paul
33 33 Denna Kelwick dkelwickw@unesco.org 1999-02-21 Employee American University of Sharjah
34 34 Eolande Earpe eearpex@woothemes.com 1993-07-17 Employee Northeast Normal University
35 35 Karleen Beinke kbeinkey@phpbb.com 1989-06-01 Employee Kyungpook National University
36 36 Morry Goodband mgoodbandz@xrea.com 1990-10-13 Student Sarhad University of Science & Information Technology, Peshawar
37 37 Buck Chappelow bchappelow10@biglobe.ne.jp 1990-11-21 Student Mohammad Ali Jinnah University
38 38 Mirabel Bunton mbunton11@stanford.edu 1992-01-03 Employee Boston University
39 39 Rosalinda Dinjes rdinjies12@ehow.com 1991-04-19 Employee Universidad de Zaragoza
40 40 Shoshana Gallelli sgallelli13@icq.com 2000-07-08 Employee Universidad Tecnológica de Panamá
41 41 Lurline Brackenbury lbrackenbury14@nsw.gov.au 2003-02-28 Employee Rochester Institute of Technology, Dubai
42 42 Vevay Itzkovwitch vitzkovwitch15.blogspot.com 1990-02-22 Employee University of Dayton
43 43 Franzen Sallans fsallans16@vinaora.com 1991-09-23 Student Sacred Heart University
44 44 Darrel Student dStudent17@tinypic.com 1995-12-12 Student Northwood University, Florida Campus
45 45 Hermie Perkins hperkins18@state.gov 2004-01-05 Student Northern Arizona University
46 46 Sahil Mangukiya sahil mangukiya@gmial.com 2002-09-20 Student DAIICT

Total rows: 46 of 46  Query complete 00:00:00.101  Ln 400, Col 1

```

Number of Tuples → 46

Query-20.

Create a new table, "discontinued_Hackathon". Create a new trigger and call after the deletion of sport from the Hackathon table. After the removal of each sport, the trigger should enter the Hack_id and Hack_name of the discontinued sport into the discontinued_Hackathon. table.

SQL Query-20.

```
create table if not exists discontinued_Hackathon(  
    Hack_id bigint,  
    Hack_Name varchar  
)  
  
create or replace function func_2()  
returns trigger  
language 'plpgsql'  
as $body$  
begin  
    insert into discontinued_Hackathon(Hack_id, Hack_Name)  
    values(old.Hack_id, old.Hack_Name);  
  
    return new;  
  
end  
  
$body$  
  
create or replace trigger trigger_2  
after delete  
on Hackathon  
for each row  
execute procedure func_2();
```

Snapshot-20.**Before Delete Operation:**

Query Query History

```
427
428 select * from Hackathon
429
```

Data output Messages Notifications

	hack_id	hack_name	name_type	date	time	duration
28	28	Vagram	Job	2023-08-27	14:13:00	3
29	29	Toughjoyfax	Internship	2022-12-25	11:17:00	4
30	30	Home Ing	Job	2023-10-22	16:20:00	3
31	31	It	Internship	2023-03-26	18:09:00	3
32	32	Alpha	Internship	2023-02-22	15:25:00	4
33	33	Pannier	Internship	2023-02-25	16:18:00	3
34	34	Zontrax	Job	2023-09-03	12:16:00	4
35	35	Gembucket	Job	2023-04-03	18:43:00	3
36	36	Redhold	Job	2023-11-01	14:58:00	3
37	37	It	Programming	2022-11-27	15:25:00	2
38	38	Tempsoft	Job	2023-01-28	15:52:00	4
39	39	Alphazap	Internship	2023-10-14	14:47:00	2
40	40	Mat Lam Tam	Programming	2023-02-23	18:57:00	2
41	41	Cardguard	Job	2023-06-20	13:04:00	4
42	42	Otcom	Job	2023-03-06	12:44:00	3
43	43	Rank	Programming	2023-02-08	13:11:00	2
44	44	Opela	Programming	2023-10-05	17:17:00	2
45	45	Wrapsafe	Job	2023-11-11	13:59:00	3

Total rows: 45 of 45 Query complete 00:00:11.066

Number of Tuples → 45**Before Delete Operation:**

Query Query History

```
427
428 select * from Hackathon
429 select * from discontinued_Hackathon
430
```

Data output Messages Notifications

	hack_id	hack_name

Total rows: 0 of 0 Query complete 00:00:05.094

Number of Tuples → 0

After Delete Operation:

delete from Hackathon where Hack_id = 10;

Select * from Hackathon;

Hackathon Data						
	hack_id [PK] bigint	hack_name character varying	name_type character varying	date date	time time without time zone	duration bigint
5	5	Y-find	Programming	2022-11-24	10:47:00	4
6	6	Alphazap	Programming	2023-07-27	10:54:00	2
7	7	Cardify	Programming	2023-01-02	14:30:00	2
8	8	Greenlam	Programming	2023-04-06	16:39:00	2
9	9	Otcom	Internship	2023-06-16	14:03:00	4
10	11	Bitwolf	Programming	2023-01-23	10:40:00	4
11	12	Konklux	Job	2023-06-09	12:56:00	2
12	13	Bitchip	Internship	2022-12-19	18:09:00	2
13	14	Stringtough	Programming	2023-03-24	17:37:00	4
14	15	Solarbreeze	Programming	2022-12-20	09:18:00	4
15	16	Keylex	Job	2022-12-09	09:19:00	3
16	17	Bytecard	Programming	2023-06-24	15:14:00	3
17	18	Sonair	Job	2023-03-26	10:06:00	4
18	19	Bigtax	Internship	2022-11-21	18:02:00	2
19	20	Sonsing	Internship	2023-02-18	09:35:00	4
20	21	Konklab	Internship	2023-05-10	15:04:00	3

Total rows: 44 of 44 Query complete 00:00:06.745

Number of Tuples → 44

After Delete Operation:

Select * from discontinued_Hackathon;

discontinued_Hackathon Data		
	hack_id bigint	hack_name character varying
1	10	Toughjoyfax

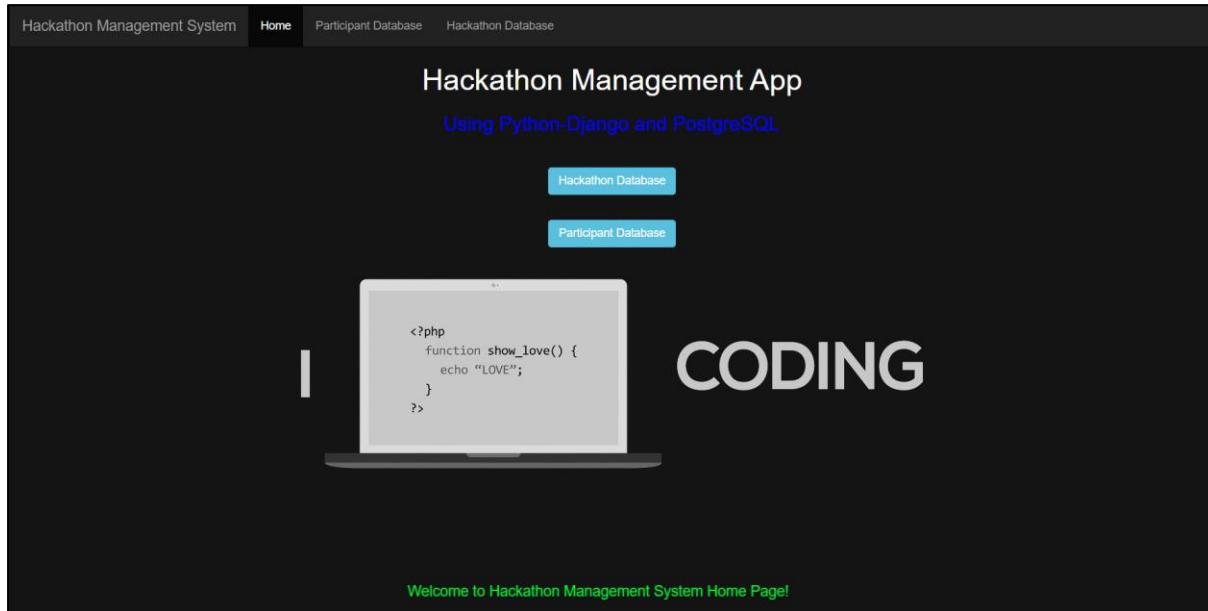
Total rows: 1 of 1 Query complete 00:00:02.560

Number of Tuples → 1

Lab 10

Screenshots of the Website that connects the Database

1. Home Page



2. Participants Database

Participant Records								
<input type="text" value="Participant ID"/> <input type="button" value="Sort"/>								
part_id	first_name	last_name	email_addr	dob	student	college	cpi	
1	Bemarr	Stegell	bstegell@hp.com	July 23, 1993	Student	Ecole Supérieure de Commerce de Toulouse	6.76	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Wakefield	Halesworth	whalesworth1@qq.com	May 25, 2004	Student	Universidad Autónoma Latinoamericana	4.31	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Jenlie	Widdowson	jwidowson2@pagesperso-orange.fr	April 23, 1995	Employee	Osaka University of Arts	9.85	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Silvia	Calvey	scalviny3@bbs.org	Dec. 30, 2002	Employee	Georg-Simon-Ohm-Fachhochschule Nürnberg	9.39	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	Monroe	Beneze	mbeneze4@imgur.com	Feb. 9, 1992	Student	Alaska Pacific University	6.35	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
6	Idette	Harback	iharback5@seattletimes.com	Oct. 15, 2000	Employee	Shanghai Institute of Foreign Trade	7.78	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
7	Sabine	Butterley	sbutterley6@earthlink.net	Dec. 24, 1998	Employee	University of Northern British Columbia	9.15	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
8	Roderich	Cissold	rcissold7@eepur.com	Nov. 23, 1987	Student	Universitas Ngurah Rai	6.9	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
9	Pearl	Hornung	phornung8@umich.edu	Jan. 4, 1987	Employee	Dankook University	6.51	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
10	Joe	Edinboro	jedinboro9@cbsc.com	March 26, 1994	Student	Beijing Petroleum University	4.87	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
11	Dorene	Disbury	ddisbury@geocities.com	Sept. 5, 1999	Employee	Latvian Academy of Music	7.54	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
12	Stamford	Hampe	shampab@ipod.com	Jan. 26, 1968	Student	Knox College	4.59	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
13	Cerissa	Antonopoulos	cantonopoulosc@cbslocal.com	Aug. 6, 2003	Employee	Rowan University	5.08	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
14	Guillbert	Basson	gbassond@kickstarter.com	July 12, 1999	Student	Ifs University College	9.57	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
15	Emilio	Eves	eevese@dagondesign.com	Oct. 28, 1988	Student	University of International Business	8.01	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Sort operation on Participants Database

➤ Sort Based on Participants CPI

The screenshot shows a web-based application titled "Hackathon Management System". The current page is "Participant Database". A search bar at the top right contains the value "CPI". Below it is a table titled "Participant Record" with a "Sort" button. The table has columns: part_id, first_name, last_name, email_addr, dob, student, college, and cpi. The rows are sorted by CPI in descending order. The last column contains "Edit" and "Delete" buttons for each row.

part_id	first_name	last_name	email_addr	dob	student	college	cpi	
28	Adelaide	Troder	atrodier@unesco.org	June 16, 1991	Employee	University of North Carolina at Wilmington	4.06	Edit Delete
35	Karleen	Beinke	kbeinkey@phppb.com	June 1, 1989	Employee	Kyungpook National University	4.23	Edit Delete
2	Wakefield	Halesworth	whalesworth1@qq.com	May 25, 2004	Student	Universidad Autónoma Latinoamericana	4.31	Edit Delete
21	Charman	Brewis	cbrewisk@360.cn	Feb. 3, 1990	Employee	American University in Cairo	4.56	Edit Delete
12	Stanford	Hampe	shampes@tripod.com	Jan. 26, 1988	Student	Knox College	4.59	Edit Delete
23	Herschel	Halpeine	hhalpeinem@uafuu.com	Feb. 2, 1995	Student	University of Asia and the Pacific	4.64	Edit Delete
10	Joe	Edinboro	jedinboro@cnnbc.com	March 26, 1994	Student	Beijing Petroleum University	4.87	Edit Delete
13	Corissa	Antonopoulos	cantonopoulos@cbslocal.com	Aug. 6, 2003	Employee	Rowan University	5.08	Edit Delete
32	Katey	Ethridge	keithridge@artisteer.com	May 21, 2002	Employee	Bethel College St. Paul	5.09	Edit Delete
19	Berni	Woodier	bwoodier@163.com	June 3, 1988	Employee	Sudan Academy of Sciences	5.27	Edit Delete
24	Gustavus	Prangie	gprangien@blogs.com	Aug. 31, 2005	Student	Tokyo University of Art and Design	5.35	Edit Delete
27	Joanie	Epsley	jepletsleyq@dsqus.com	Jan. 10, 1992	Employee	California State University, Fresno	5.52	Edit Delete
31	Godfree	Muriel	gmurielu@artisteer.com	July 26, 1992	Student	Universidad Pontificia Comillas	5.54	Edit Delete
25	Heall	Myers	hmyerso@multiply.com	Jan. 8, 1989	Student	Galilee College	5.96	Edit Delete

Number of Tuples ➔ 45

Insert operation on Participants Database

➤ Before Insert:

The screenshot shows the same "Participant Database" page as the previous one, but with a different set of data. The table now contains 45 rows of participant information, starting from part_id 29. The columns and structure are identical to the previous screenshot, showing fields like first_name, last_name, email_addr, dob, student status, college, and CPI.

29	Francis	Gery	fgerys@imdb.com	Jan. 1, 1988	Student	Universidad Tecnológica del Perú	8.08	Edit Delete
30	Antoinette	Perigeaux	aperigeaux@yelp.com	March 16, 1997	Employee	Universidad Agraria del Ecuador	9.48	Edit Delete
31	Godfree	Muriel	gmurielu@artisteer.com	July 26, 1992	Student	Universidad Pontificia Comillas	5.54	Edit Delete
32	Katey	Ethridge	keithridge@artisteer.com	May 21, 2002	Employee	Bethel College St. Paul	5.09	Edit Delete
33	Denna	Ketwick	dketwick@unesco.org	Feb. 21, 1999	Employee	American University of Sharjah	8.94	Edit Delete
34	Eolandie	Earpie	earapex@woothemes.com	July 17, 1993	Employee	Northeast Normal University	7.04	Edit Delete
35	Karleen	Beinke	kbeinkey@phppb.com	June 1, 1989	Employee	Kyungpook National University	4.23	Edit Delete
36	Morry	Goodband	mgoodbandz@xrea.com	Oct. 13, 1990	Student	Sarhad University of Science & Information Technology, Peshawar	9.42	Edit Delete
37	Buck	Chappelow	bchappelow10@biglobe.ne.jp	Nov. 21, 1990	Student	Mohammad Ali Jinnah University	8.47	Edit Delete
38	Mirabel	Bunton	mbunton11@stanford.edu	Jan. 3, 1992	Employee	Boston University	7.44	Edit Delete
39	Rosalinda	Dinges	rdinges12@ehow.com	April 19, 1991	Employee	Universidad de Zaragoza	7.47	Edit Delete
40	Shoshana	Gallelli	sgallelli13@icq.com	July 8, 2000	Employee	Universidad Tecnológica de Panamá	6.76	Edit Delete
41	Lurline	Brackenbury	lbrackenbury14@new.gov.au	Feb. 28, 2003	Employee	Rochester Institute of Technology, Dubai	8.88	Edit Delete
42	Vevay	Itzkowitz	vitzkowitz15@blogspot.com	Feb. 22, 1990	Employee	University of Dayton	9.17	Edit Delete
43	Franzen	Sallans	fsallans16@vinaora.com	Sept. 23, 1991	Student	Sacred Heart University	6.84	Edit Delete
44	Darrel	Student	dStudent17@impc.com	Dec. 12, 1995	Student	Northwood University, Florida Campus	7.35	Edit Delete
45	Hermie	Perkins	hperkins18@state.gov	Jan. 5, 2004	Student	Northern Arizona University	6.56	Edit Delete

Number of Tuples ➔ 45

➤ Insert Participants Data

Hackathon Management System Home Participant Database Hackathon Database

Insert Participant Details

part_id	202001251
first_name	Sahil
last_name	Mangukuya
email_add	202001251@dalict.ac.in
dob	20-09-2002
student	Student
college	DAIICT
cpi	7.22
<input type="button" value="Submit"/>	

[back](#)

➤ After Insert

33	Denna	Kelwick	dkelwickw@unesco.org	Feb 21, 1999	Employee	American University of Sharjah	8.94	Edit	Delete
34	Eolande	Earpe	earpex@woothemes.com	July 17, 1993	Employee	Northeast Normal University	7.04	Edit	Delete
35	Karleen	Beinke	kbeinkey@phpbb.com	June 1, 1989	Employee	Kyungpook National University	4.23	Edit	Delete
36	Morry	Goodband	mgoodbandz@xrea.com	Oct. 13, 1990	Student	Sarhad University of Science & Information Technology, Peshawar	9.42	Edit	Delete
37	Buck	Chappelow	bchappelow10@biglobe.ne.jp	Nov 21, 1990	Student	Mohammad Ali Jinnah University	8.47	Edit	Delete
38	Mirabel	Bunton	mbunton11@stanford.edu	Jan. 3, 1992	Employee	Boston University	7.44	Edit	Delete
39	Rosalinda	Dinjes	rdinjes12@ehow.com	April 19, 1991	Employee	Universidad de Zaragoza	7.47	Edit	Delete
40	Shoshana	Gallelli	sgallelli13@icq.com	July 8, 2000	Employee	Universidad Tecnológica de Panamá	6.76	Edit	Delete
41	Lurline	Bräckenbury	lbrackenbury14@nsw.gov.au	Feb. 28, 2003	Employee	Rochester Institute of Technology, Dubai	8.88	Edit	Delete
42	Vevay	Itzkowitch	vitzkovitch15@blogspot.com	Feb. 22, 1990	Employee	University of Dayton	9.17	Edit	Delete
43	Franzen	Sallans	fsallans16@vinaora.com	Sept. 23, 1991	Student	Sacred Heart University	6.84	Edit	Delete
44	Darrel	Student	dStudent17@tinypic.com	Dec. 12, 1995	Student	Northwood University, Florida Campus	7.35	Edit	Delete
45	Hermie	Perkins	hperkins18@state.gov	Jan. 5, 2004	Student	Northern Arizona University	6.56	Edit	Delete
202001251	Sahil	Mangukuya	202001251@dalict.ac.in	Nov. 4, 2022	Student	DAIICT	7.22	Edit	Delete

[Add Participant](#)

Number of Tuples ➔ 46

Update operation on Participants Database

➤ Before Update

Participant Records								
				Participant ID	Sort			
part_id	first_name	last_name	email_add	dob	student	college	cpi	
1	Bernarr	Stegell	bstege10@hp.com	July 23, 1993	Student	Ecole Supérieure de Commerce de Toulouse	6.76	Edit Delete
2	Wakefield	Halesworth	whalesworth1@qq.com	May 25, 2004	Student	Universidad Autónoma Latinoamericana	4.31	Edit Delete
3	Jerrie	Widdowson	jwiddowson2@pagesperso-orange.fr	April 23, 1995	Employee	Osaka University of Arts	9.85	Edit Delete
4	Silvia	Calvey	scalvey3@pbs.org	Dec. 30, 2002	Employee	Georg-Simon-Ohm-Fachhochschule Nürnberg	9.39	Edit Delete
5	Monroe	Beneze	mbeneze4@imgur.com	Feb. 9, 1992	Student	Alaska Pacific University	6.35	Edit Delete
6	Idette	Harback	iharback5@seattletimes.com	Oct. 15, 2000	Employee	Shanghai Institute of Foreign Trade	7.78	Edit Delete
7	Sabine	Butterley	sbutterley6@earthlink.net	Dec. 24, 1998	Employee	University of Northern British Columbia	9.15	Edit Delete
8	Roderich	Clissold	rclissold7@eepurl.com	Nov. 23, 1987	Student	Universitas Ngurah Rai	6.9	Edit Delete
9	Pearl	Hornung	phornung8@umich.edu	Jan. 4, 1987	Employee	Dankook University	6.51	Edit Delete
10	Joe	Edinboro	jedinboro9@cnbc.com	March 26, 1994	Student	Beijing Petroleum University	4.87	Edit Delete
11	Dorene	Disbury	ddisbury@qecities.com	Sept. 5,	Employee	Latvian Academy of Music	7.54	Edit Delete

Number of Tuples → 45

Update First_Name, DOB and CPI of Part_id = 8;

➤ Update Participants Data

Update Record in Participant

part_id	<input type="text" value="8"/>
first_name	<input type="text" value="Manan"/>
last_name	<input type="text" value="Clissold"/>
email_add	<input type="text" value="rclissold7@eepurl.com"/>
dob	<input type="text" value="dd-mm-yyyy"/>
student	<input type="text" value="Student"/>
college	<input type="text" value="Universitas Ngurah Rai"/>
cpi	<input type="text" value="9.02"/>
Update	Record Updated Successfully..!

[back](#)

➤ After Update

34	Eolande	Earpe	earpex@woothemes.com	July 17, 1993	Employee	Northeast Normal University	7.04	Edit	Delete
35	Karleen	Beinke	kbeinkey@phpbb.com	June 1, 1989	Employee	Kyungpook National University	4.23	Edit	Delete
36	Morry	Goodband	mgoodbandz@xrea.com	Oct. 13, 1990	Student	Sarhad University of Science & Information Technology, Peshawar	9.42	Edit	Delete
37	Buck	Chappelow	bchappelow10@biglobe.ne.jp	Nov. 21, 1990	Student	Mohammad Ali Jinnah University	8.47	Edit	Delete
38	Mirabel	Bunton	mbunton11@stanford.edu	Jan. 3, 1992	Employee	Boston University	7.44	Edit	Delete
39	Rosalinda	Dinjes	rdinjes12@ehow.com	April 19, 1991	Employee	Universidad de Zaragoza	7.47	Edit	Delete
40	Shoshana	Gallelli	sgallelli13@icq.com	July 8, 2000	Employee	Universidad Tecnológica de Panamá	6.76	Edit	Delete
41	Lurline	Brackenbury	lbrackenbury14@nsw.gov.au	Feb. 28, 2003	Employee	Rochester Institute of Technology, Dubai	8.88	Edit	Delete
42	Vevay	Itzkovwitch	vitzkovwitch15.blogspot.com	Feb. 22, 1990	Employee	University of Dayton	9.17	Edit	Delete
43	Franzen	Sallans	fsallans16@vinaora.com	Sept. 23, 1991	Student	Sacred Heart University	6.84	Edit	Delete
44	Darrel	Student	dStudent17@tinypic.com	Dec. 12, 1995	Student	Northwood University, Florida Campus	7.35	Edit	Delete
45	Hermie	Perkins	hperkins18@state.gov	Jan. 5, 2004	Student	Northern Arizona University	6.56	Edit	Delete
8	Manan	Clissold	rclissold7@eepurl.com	June 18, 2009	Student	Universitas Ngurah Rai	9.02	Edit	Delete

Add Participant

Number of Tuples ➔ 45

Delete operation on Participants Database

➤ Before Delete

36	Morry	Goodband	mgoodbandz@xrea.com	Oct. 13, 1990	Student	Sarhad University of Science & Information Technology, Peshawar	9.42	Edit	Delete
37	Buck	Chappelow	bchappelow10@biglobe.ne.jp	Nov. 21, 1990	Student	Mohammad Ali Jinnah University	8.47	Edit	Delete
38	Mirabel	Bunton	mbunton11@stanford.edu	Jan. 3, 1992	Employee	Boston University	7.44	Edit	Delete
39	Rosalinda	Dinjes	rdinjes12@ehow.com	April 19, 1991	Employee	Universidad de Zaragoza	7.47	Edit	Delete
40	Shoshana	Gallelli	sgallelli13@icq.com	July 8, 2000	Employee	Universidad Tecnológica de Panamá	6.76	Edit	Delete
41	Lurline	Brackenbury	lbrackenbury14@nsw.gov.au	Feb. 28, 2003	Employee	Rochester Institute of Technology, Dubai	8.88	Edit	Delete
42	Vevay	Itzkovwitch	vitzkovwitch15.blogspot.com	Feb. 22, 1990	Employee	University of Dayton	9.17	Edit	Delete
43	Franzen	Sallans	fsallans16@vinaora.com	Sept. 23, 1991	Student	Sacred Heart University	6.84	Edit	Delete
44	Darrel	Student	dStudent17@tinypic.com	Dec. 12, 1995	Student	Northwood University, Florida Campus	7.35	Edit	Delete
45	Hermie	Perkins	hperkins18@state.gov	Jan. 5, 2004	Student	Northern Arizona University	6.56	Edit	Delete
8	Manan	Clissold	rclissold7@eepurl.com	June 18, 2009	Student	Universitas Ngurah Rai	9.02	Edit	Delete

Add Participant

Number of Tuples ➔ 46

Delete a Record of Part_id = 202001251.

➤ Asking For Confirmation

				127.0.0.1:8000 says						
	36	Morry	Goodband	Are you sure you want to delete the record ?				of Science & Information	9.42	Edit Delete
	37	Buck	Chappelow	bchappelow10@biglobe.ne.jp	mbunton11@stanford.edu	Jan. 3, 1990	Employee	Jawar	8.47	Edit Delete
	38	Mirabel	Bunton	mbunton11@stanford.edu	rdinjes12@ehow.com	April 19, 1991	Employee	Boston University	7.44	Edit Delete
	39	Rosalinda	Dinjes	rdinjes12@ehow.com	sgallelli13@icq.com	July 8, 2000	Employee	Universidad de Zaragoza	7.47	Edit Delete
	40	Shoshana	Gallelli	sgallelli13@icq.com	lbrackenbury14@nsw.gov.au	Feb. 28, 2003	Employee	Universidad Tecnológica de Panamá	6.76	Edit Delete
	41	Lurline	Brackenbury	lbrackenbury14@nsw.gov.au	vitzkovwitch15.blogspot.com	Feb. 22, 1990	Employee	Rochester Institute of Technology, Dubai	8.88	Edit Delete
	42	Vevay	Itzkovwitch	vitzkovwitch15.blogspot.com	fsallans16@vinaora.com	Sept. 23, 1991	Employee	University of Dayton	9.17	Edit Delete
	43	Franzen	Sallans	fsallans16@vinaora.com	dStudent17@tinypic.com	Dec. 12, 1995	Student	Sacred Heart University	6.84	Edit Delete
	44	Darrel	Student	dStudent17@tinypic.com	Manan	June 18, 2009	Student	Northwood University, Florida Campus	7.35	Edit Delete
	45	Hermie	Perkins	hperkins18@state.gov	Clissold	Jan. 5, 2004	Student	Northern Arizona University	6.56	Edit Delete
	8	Manan	Clissold	rclissold7@eepurl.com	202001251@daiict.ac.in	June 18, 2009	Student	Universitas Ngurah Rai	9.02	Edit Delete
	202001251	Sahil	Mangukya	202001251@daiict.ac.in		Sept. 20, 2003	Student	DAIICT	7.22	Edit Delete
								Add Participant		

➤ After Delete

Part_id = 202001251 was deleted.										
	34	Eolande	Earpe	earpex@woothemes.com	July 17, 1993	Employee	Northeast Normal University		7.04	Edit Delete
	35	Karleen	Beinke	kbeinkey@phpbb.com	June 1, 1989	Employee	Kyungpook National University		4.23	Edit Delete
	36	Morry	Goodband	mgoodbandz@xrea.com	Oct. 13, 1990	Student	Sarhad University of Science & Information Technology, Peshawar		9.42	Edit Delete
	37	Buck	Chappelow	bchappelow10@biglobe.ne.jp	Nov. 21, 1990	Student	Mohammad Ali Jinnah University		8.47	Edit Delete
	38	Mirabel	Bunton	mbunton11@stanford.edu	Jan. 3, 1992	Employee	Boston University		7.44	Edit Delete
	39	Rosalinda	Dinjes	rdinjes12@ehow.com	April 19, 1991	Employee	Universidad de Zaragoza		7.47	Edit Delete
	40	Shoshana	Gallelli	sgallelli13@icq.com	July 8, 2000	Employee	Universidad Tecnológica de Panamá		6.76	Edit Delete
	41	Lurline	Brackenbury	lbrackenbury14@nsw.gov.au	Feb. 28, 2003	Employee	Rochester Institute of Technology, Dubai		8.88	Edit Delete
	42	Vevay	Itzkovwitch	vitzkovwitch15.blogspot.com	Feb. 22, 1990	Employee	University of Dayton		9.17	Edit Delete
	43	Franzen	Sallans	fsallans16@vinaora.com	Sept. 23, 1991	Student	Sacred Heart University		6.84	Edit Delete
	44	Darrel	Student	dStudent17@tinypic.com	Manan	Dec. 12, 1995	Student	Northwood University, Florida Campus	7.35	Edit Delete
	45	Hermie	Perkins	hperkins18@state.gov	Clissold	Jan. 5, 2004	Student	Northern Arizona University	6.56	Edit Delete
	8	Manan	Clissold	rclissold7@eepurl.com	202001251@daiict.ac.in	June 18, 2009	Student	Universitas Ngurah Rai	9.02	Edit Delete
						Add Participant				

Number of Tuples ➔ 45

3. Hackathon Database

Hackathon Records							
		Type	date	time	duration		
Hackathon ID	Hackathon Name					Edit	Delete
1	Fixflex	Programming	Oct. 3, 2023	7:19 p.m.	4	Edit	Delete
2	Stronghold	Job	June 29, 2023	11:32 a.m.	2	Edit	Delete
3	Namfix	Job	June 18, 2023	12:55 p.m.	2	Edit	Delete
4	Fintone	Programming	Dec. 14, 2022	12:40 p.m.	4	Edit	Delete
5	Y-find	Programming	Nov. 24, 2022	10:47 a.m.	4	Edit	Delete
6	Alphazap	Programming	July 27, 2023	10:54 a.m.	2	Edit	Delete
7	Cardify	Programming	Jan. 2, 2023	2:30 p.m.	2	Edit	Delete
8	Greenlam	Programming	April 6, 2023	4:39 p.m.	2	Edit	Delete
9	Otcom	Internship	June 16, 2023	2:03 p.m.	4	Edit	Delete
10	Toughjoyfax	Job	Aug. 12, 2023	1 p.m.	2	Edit	Delete
11	Bitwolf	Programming	Jan. 23, 2023	10:40 a.m.	4	Edit	Delete
12	Konklux	Job	June 9, 2023	12:56 p.m.	2	Edit	Delete
13	Bitchip	Internship	Dec. 19, 2022	6:09 p.m.	2	Edit	Delete
14	Stringtough	Programming	March 24, 2023	5:37 p.m.	4	Edit	Delete
15	Solarbreeze	Programming	Dec. 20, 2022	9:18 a.m.	4	Edit	Delete

Sort operation on Hackathon Database

➤ Sort Based on Hackathon Date

Hackathon Record						
		Type	date	time	duration	
Hackathon ID	Hackathon Name					
27	Alphazap	Job	Nov. 21, 2022	6:07 p.m.	2	Edit Delete
19	Bigtax	Internship	Nov. 21, 2022	6:02 p.m.	2	Edit Delete
5	Y-find	Programming	Nov. 24, 2022	10:47 a.m.	4	Edit Delete
37	It	Programming	Nov. 27, 2022	3:25 p.m.	2	Edit Delete
16	Keylex	Job	Dec. 9, 2022	9:19 a.m.	3	Edit Delete
4	Fintone	Programming	Dec. 14, 2022	12:40 p.m.	4	Edit Delete
13	Bitchip	Internship	Dec. 19, 2022	6:09 p.m.	2	Edit Delete
15	Solarbreeze	Programming	Dec. 20, 2022	9:18 a.m.	4	Edit Delete
29	Toughjoyfax	Internship	Dec. 25, 2022	11:17 a.m.	4	Edit Delete
7	Cardify	Programming	Jan. 2, 2023	2:30 p.m.	2	Edit Delete
23	Wrapsafe	Internship	Jan. 5, 2023	2:36 p.m.	2	Edit Delete
11	Bitwolf	Programming	Jan. 23, 2023	10:40 a.m.	4	Edit Delete
38	Tempsoft	Job	Jan. 28, 2023	3:52 p.m.	4	Edit Delete
43	Rank	Programming	Feb. 8, 2023	1:11 p.m.	2	Edit Delete
22	Asoka	Programming	Feb. 17, 2023	7:07 p.m.	3	Edit Delete

Number of Tuples ➔ 45

Insert operation on Hackathon Database

➤ **Before Insert:**

25	Kanlam	Internship	Oct. 22, 2023	7:20 p.m.	3	Edit	Delete	
26	Stringtough	Internship	April 1, 2023	2:29 p.m.	2	Edit	Delete	
27	Alphazap	Job	Nov. 21, 2022	6:07 p.m.	2	Edit	Delete	
28	Vagram	Job	Aug. 27, 2023	2:13 p.m.	3	Edit	Delete	
29	Toughjoyfax	Internship	Dec. 25, 2022	11:17 a.m.	4	Edit	Delete	
30	Home Ing	Job	Oct. 22, 2023	4:20 p.m.	3	Edit	Delete	
31	It	Internship	March 26, 2023	6:09 p.m.	3	Edit	Delete	
32	Alpha	Internship	Feb. 22, 2023	3:25 p.m.	4	Edit	Delete	
33	Pannier	Internship	Feb. 25, 2023	4:18 p.m.	3	Edit	Delete	
34	Zontrax	Job	Sept. 3, 2023	12:16 p.m.	4	Edit	Delete	
35	Gembucket	Job	April 3, 2023	6:43 p.m.	3	Edit	Delete	
36	Redhold	Job	Nov. 1, 2023	2:58 p.m.	3	Edit	Delete	
37	It	Programming	Nov. 27, 2022	3:25 p.m.	2	Edit	Delete	
38	Tempsoft	Job	Jan. 28, 2023	3:52 p.m.	4	Edit	Delete	
39	Alphazap	Internship	Oct. 14, 2023	2:47 p.m.	2	Edit	Delete	
40	Mat Lam Tam	Programming	Feb. 23, 2023	6:57 p.m.	2	Edit	Delete	
41	Cardguard	Job	June 20, 2023	1:04 p.m.	4	Edit	Delete	
42	Otcom	Job	March 6, 2023	12:44 p.m.	3	Edit	Delete	
43	Rank	Programming	Feb. 8, 2023	1:11 p.m.	2	Edit	Delete	
44	Opela	Programming	Oct. 5, 2023	5:17 p.m.	2	Edit	Delete	
45	Wrapsafe	Job	Nov. 11, 2023	1:59 p.m.	3	Edit	Delete	

back

Number of Tuples ➔ 45

➤ **Insert Hackathon Data**

Hackathon Management System
Home
Participant Database
Hackathon Database

Insert Hackathon Details

hack_id	<input type="text" value="46"/>
hack_name	<input type="text" value="Google Summer of Code"/>
name_type	<input type="text" value="Prize"/>
date	<input type="text" value="15-05-2023"/>
time	<input type="text" value="09:30"/>
duration	<input type="text" value="4"/>
<input type="button" value="Submit"/> back	

➤ After Insert



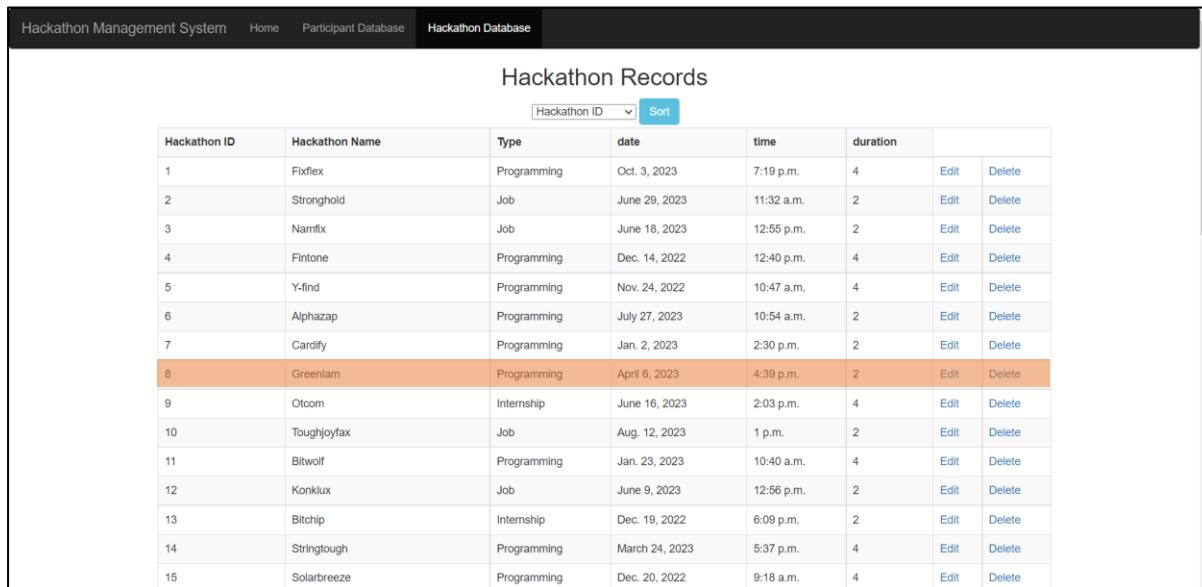
A screenshot of a database table titled "Hackathon Database". The table has columns: ID, Name, Type, Date, Time, Duration, Edit, and Delete. There are 46 rows of data. The last row, which contains the value "Google Summer of Code" in the Name column and "Prize" in the Type column, is highlighted with an orange background.

26	Stringtough	Internship	April 1, 2023	2:29 p.m.	2	Edit	Delete
27	Alphazap	Job	Nov 21, 2022	6:07 p.m.	2	Edit	Delete
28	Vagram	Job	Aug 27, 2023	2:13 p.m.	3	Edit	Delete
29	Toughjoyfax	Internship	Dec 25, 2022	11:17 a.m.	4	Edit	Delete
30	Home Ing	Job	Oct 22, 2023	4:20 p.m.	3	Edit	Delete
31	It	Internship	March 26, 2023	6:09 p.m.	3	Edit	Delete
32	Alpha	Internship	Feb. 22, 2023	3:25 p.m.	4	Edit	Delete
33	Pannier	Internship	Feb. 25, 2023	4:18 p.m.	3	Edit	Delete
34	Zontrax	Job	Sept. 3, 2023	12:16 p.m.	4	Edit	Delete
35	Gembucket	Job	April 3, 2023	6:43 p.m.	3	Edit	Delete
36	Redhold	Job	Nov. 1, 2023	2:58 p.m.	3	Edit	Delete
37	It	Programming	Nov. 27, 2022	3:25 p.m.	2	Edit	Delete
38	Tempsoft	Job	Jan. 28, 2023	3:52 p.m.	4	Edit	Delete
39	Alphazap	Internship	Oct. 14, 2023	2:47 p.m.	2	Edit	Delete
40	Mat Lam Tam	Programming	Feb. 23, 2023	6:57 p.m.	2	Edit	Delete
41	Cardguard	Job	June 20, 2023	1:04 p.m.	4	Edit	Delete
42	Otcom	Job	March 6, 2023	12:44 p.m.	3	Edit	Delete
43	Rank	Programming	Feb. 8, 2023	1:11 p.m.	2	Edit	Delete
44	Opela	Programming	Oct. 5, 2023	5:17 p.m.	2	Edit	Delete
45	Wrapsafe	Job	Nov. 11, 2023	1:59 p.m.	3	Edit	Delete
46	Google Summer of Code	Prize	May 15, 2023	9:30 a.m.	4	Edit	Delete

Number of Tuples ➔ 46

Update operation on Hackathon Database

➤ Before Update



A screenshot of a database table titled "Hackathon Records". The table has columns: Hackathon ID, Hackathon Name, Type, Date, Time, Duration, Edit, and Delete. There are 15 rows of data. The last row, which contains the value "Solarbreeze" in the Name column and "Programming" in the Type column, is highlighted with an orange background.

Hackathon ID	Hackathon Name	Type	Date	Time	Duration		
1	Fixflex	Programming	Oct. 3, 2023	7:19 p.m.	4	Edit	Delete
2	Stronghold	Job	June 29, 2023	11:32 a.m.	2	Edit	Delete
3	Namfix	Job	June 18, 2023	12:55 p.m.	2	Edit	Delete
4	Fintone	Programming	Dec. 14, 2022	12:40 p.m.	4	Edit	Delete
5	Y-find	Programming	Nov. 24, 2022	10:47 a.m.	4	Edit	Delete
6	Alphazap	Programming	July 27, 2023	10:54 a.m.	2	Edit	Delete
7	Cardify	Programming	Jan. 2, 2023	2:30 p.m.	2	Edit	Delete
8	Greenlam	Programming	April 6, 2023	4:39 p.m.	2	Edit	Delete
9	Otcom	Internship	June 16, 2023	2:03 p.m.	4	Edit	Delete
10	Toughjoyfax	Job	Aug. 12, 2023	1 p.m.	2	Edit	Delete
11	Bitwolf	Programming	Jan. 23, 2023	10:40 a.m.	4	Edit	Delete
12	Konklux	Job	June 9, 2023	12:56 p.m.	2	Edit	Delete
13	Bitchip	Internship	Dec. 19, 2022	6:09 p.m.	2	Edit	Delete
14	Stringtough	Programming	March 24, 2023	5:37 p.m.	4	Edit	Delete
15	Solarbreeze	Programming	Dec. 20, 2022	9:18 a.m.	4	Edit	Delete

Number of Tuples ➔ 46

Update Hackathon Name, Name Type and Duration of Hack_id = 8;

➤ Update Hackathon Data

Hackathon Management System Home Participant Database Hackathon Database

Update Record in Hackathon

hack_id	8
hack_name	Smart India Hackathon
name_type	Job
date	06-04-2023
time	16:39
duration	3

Update **back**

➤ After Update

27	Alphazap	Job	Nov. 21, 2022	6:07 p.m.	2	Edit	Delete
28	Vagram	Job	Aug. 27, 2023	2:13 p.m.	3	Edit	Delete
29	Toughjoyfax	Internship	Dec. 25, 2022	11:17 a.m.	4	Edit	Delete
30	Home Ing	Job	Oct. 22, 2023	4:20 p.m.	3	Edit	Delete
31	It	Internship	March 26, 2023	6:09 p.m.	3	Edit	Delete
32	Alpha	Internship	Feb. 22, 2023	3:25 p.m.	4	Edit	Delete
33	Pannier	Internship	Feb. 25, 2023	4:18 p.m.	3	Edit	Delete
34	Zontrax	Job	Sept. 3, 2023	12:16 p.m.	4	Edit	Delete
35	Gembucket	Job	April 3, 2023	6:43 p.m.	3	Edit	Delete
36	Redhold	Job	Nov. 1, 2023	2:58 p.m.	3	Edit	Delete
37	It	Programming	Nov. 27, 2022	3:25 p.m.	2	Edit	Delete
38	Tempsoft	Job	Jan. 28, 2023	3:52 p.m.	4	Edit	Delete
39	Alphazap	Internship	Oct. 14, 2023	2:47 p.m.	2	Edit	Delete
40	Mat Lam Tam	Programming	Feb. 23, 2023	6:57 p.m.	2	Edit	Delete
41	Cardguard	Job	June 20, 2023	1:04 p.m.	4	Edit	Delete
42	Otcom	Job	March 6, 2023	12:44 p.m.	3	Edit	Delete
43	Rank	Programming	Feb. 8, 2023	1:11 p.m.	2	Edit	Delete
44	Opela	Programming	Oct. 5, 2023	5:17 p.m.	2	Edit	Delete
45	Wrapsafe	Job	Nov. 11, 2023	1:59 p.m.	3	Edit	Delete
46	Google Summer of Code	Prize	May 15, 2023	9:30 a.m.	4	Edit	Delete
8	Smart India Hackathon	Job	April 6, 2023	4:39 p.m.	3	Edit	Delete

Add Hackathon

Number of Tuples ➔ 46

Delete operation on Hackathon Database

➤ Before Delete

27	Alphazap	Job	Nov 21, 2022	6:07 p.m.	2	Edit	Delete
28	Vagram	Job	Aug 27, 2023	2:13 p.m.	3	Edit	Delete
29	Toughjoyfax	Internship	Dec 25, 2022	11:17 a.m.	4	Edit	Delete
30	Home Ing	Job	Oct 22, 2023	4:20 p.m.	3	Edit	Delete
31	It	Internship	March 26, 2023	6:09 p.m.	3	Edit	Delete
32	Alpha	Internship	Feb. 22, 2023	3:25 p.m.	4	Edit	Delete
33	Pannier	Internship	Feb. 25, 2023	4:18 p.m.	3	Edit	Delete
34	Zontrax	Job	Sept. 3, 2023	12:16 p.m.	4	Edit	Delete
35	Gembucket	Job	April 3, 2023	6:43 p.m.	3	Edit	Delete
36	Redhold	Job	Nov. 1, 2023	2:58 p.m.	3	Edit	Delete
37	It	Programming	Nov. 27, 2022	3:25 p.m.	2	Edit	Delete
38	Tempsoft	Job	Jan. 28, 2023	3:52 p.m.	4	Edit	Delete
39	Alphazap	Internship	Oct. 14, 2023	2:47 p.m.	2	Edit	Delete
40	Mat Lam Tam	Programming	Feb. 23, 2023	6:57 p.m.	2	Edit	Delete
41	Cardguard	Job	June 20, 2023	1:04 p.m.	4	Edit	Delete
42	Otcom	Job	March 6, 2023	12:44 p.m.	3	Edit	Delete
43	Rank	Programming	Feb. 8, 2023	1:11 p.m.	2	Edit	Delete
44	Opela	Programming	Oct. 5, 2023	5:17 p.m.	2	Edit	Delete
45	Wrapsafe	Job	Nov. 11, 2023	1:59 p.m.	3	Edit	Delete
46	Google Summer of Code	Prize	May 15, 2023	9:30 a.m.	4	Edit	Delete
8	Smart India Hackathon	Job	April 6, 2023	4:39 p.m.	3	Edit	Delete

[Add Hackathon](#)

Number of Tuples ➔ 46

Delete a Record of Hack_id = 46.

➤ Asking For Confirmation

27	Alphazap	127.0.0.1:8000 says Are you sure you want to delete the record ?			2	Edit	Delete
28	Vagram				3	Edit	Delete
29	Toughjoyfax				4	Edit	Delete
30	Home Ing				3	Edit	Delete
31	It	Internship	March 26, 2023	6:09 p.m.	3	Edit	Delete
32	Alpha	Internship	Feb. 22, 2023	3:25 p.m.	4	Edit	Delete
33	Pannier	Internship	Feb. 25, 2023	4:18 p.m.	3	Edit	Delete
34	Zontrax	Job	Sept. 3, 2023	12:16 p.m.	4	Edit	Delete
35	Gembucket	Job	April 3, 2023	6:43 p.m.	3	Edit	Delete
36	Redhold	Job	Nov. 1, 2023	2:58 p.m.	3	Edit	Delete
37	It	Programming	Nov. 27, 2022	3:25 p.m.	2	Edit	Delete
38	Tempsoft	Job	Jan. 28, 2023	3:52 p.m.	4	Edit	Delete
39	Alphazap	Internship	Oct. 14, 2023	2:47 p.m.	2	Edit	Delete
40	Mat Lam Tam	Programming	Feb. 23, 2023	6:57 p.m.	2	Edit	Delete
41	Cardguard	Job	June 20, 2023	1:04 p.m.	4	Edit	Delete
42	Otcom	Job	March 6, 2023	12:44 p.m.	3	Edit	Delete
43	Rank	Programming	Feb. 8, 2023	1:11 p.m.	2	Edit	Delete
44	Opela	Programming	Oct. 5, 2023	5:17 p.m.	2	Edit	Delete
45	Wrapsafe	Job	Nov. 11, 2023	1:59 p.m.	3	Edit	Delete
46	Google Summer of Code	Prize	May 15, 2023	9:30 a.m.	4	Edit	Delete
8	Smart India Hackathon	Job	April 6, 2023	4:39 p.m.	3	Edit	Delete

127.0.0.1:8000/Delete2/46 [Add Hackathon](#)

➤ After Delete

26	Stringtough	Internship	April 1, 2023	2:29 p.m.	2	Edit	Delete
27	Alphazap	Job	Nov 21, 2022	6:07 p.m.	2	Edit	Delete
28	Vagram	Job	Aug 27, 2023	2:13 p.m.	3	Edit	Delete
29	Toughjyfax	Internship	Dec. 25, 2022	11:17 a.m.	4	Edit	Delete
30	Home Ing	Job	Oct 22, 2023	4:20 p.m.	3	Edit	Delete
31	It	Internship	March 26, 2023	6:09 p.m.	3	Edit	Delete
32	Alpha	Internship	Feb. 22, 2023	3:25 p.m.	4	Edit	Delete
33	Pannier	Internship	Feb. 25, 2023	4:18 p.m.	3	Edit	Delete
34	Zontrax	Job	Sept. 3, 2023	12:16 p.m.	4	Edit	Delete
35	Gembucket	Job	April 3, 2023	6:43 p.m.	3	Edit	Delete
36	Redhold	Job	Nov. 1, 2023	2:58 p.m.	3	Edit	Delete
37	It	Programming	Nov. 27, 2022	3:25 p.m.	2	Edit	Delete
38	Tempsoft	Job	Jan. 28, 2023	3:52 p.m.	4	Edit	Delete
39	Alphazap	Internship	Oct. 14, 2023	2:47 p.m.	2	Edit	Delete
40	Mat Lam Tam	Programming	Feb. 23, 2023	6:57 p.m.	2	Edit	Delete
41	Cardguard	Job	June 20, 2023	1:04 p.m.	4	Edit	Delete
42	Otcom	Job	March 6, 2023	12:44 p.m.	3	Edit	Delete
43	Rank	Programming	Feb. 8, 2023	1:11 p.m.	2	Edit	Delete
44	Opela	Programming	Oct. 5, 2023	5:17 p.m.	2	Edit	Delete
45	Wrapsafe	Job	Nov. 11, 2023	1:59 p.m.	3	Edit	Delete
8	Smart India Hackathon	Job	April 6, 2023	4:39 p.m.	3	Edit	Delete

Hack_id = 46
was deleted.

Add Hackathon

Number of Tuples ➔ 45

Run Query operation on Participant Database

SQL Query-1.

Select * from participant where cpi between 7.2 and 8.6 order by cpi;

Snapshot-1.

➤ Output in PostgreSQL

The screenshot shows a PostgreSQL query editor interface. The query entered is:

```
30
31 select * from participant where cpi BETWEEN 7.2 and 8.6 order by cpi;
32
```

The results table has the following columns:

	part_id [PK] bigint	first_name character varying	last_name character varying	email_add character varying	dob date	student character varying	college character varying	cpi double precision
1	44	Darrel	Student	dStudent17@tinypic.com	1995-12-12	Student	Northwood University, Florida Ca...	7.35
2	38	Mirabel	Bunton	mbunton11@stanford.edu	1992-01-03	Employee	Boston University	7.44
3	18	Kally	Radnedge	kradnedgeh@engadget.com	1991-06-12	Employee	Chongqing Wenli University	7.46
4	39	Rosalinda	Dinjes	rdinjes12@ehow.com	1991-04-19	Employee	Universidad de Zaragoza	7.47
5	11	Dorene	Disbury	ddisbursya@geocities.com	1999-09-05	Employee	Latvian Academy of Music	7.54
6	22	Sabra	Symms	ssy whole ms@360.cn	1998-06-08	Employee	Lenoir-Rhyne College	7.62
7	6	Idette	Harback	iharback5@seattletimes.com	2000-10-15	Employee	Shanghai Institute of Foreign Trade	7.78
8	17	Jozef	Matis	jmati sg@bbb.org	1986-02-04	Student	Fairfield University	7.99
9	15	Emilio	Eves	eevese@dagondesign.com	1988-10-28	Student	University of International Business...	8.01
10	29	Francis	Gery	fgerys@imdb.com	1988-01-01	Student	Universidad Tecnológica del Perú	8.08
11	20	Christye	Hairy es	chairy es@elegantthemes.co...	1998-09-28	Employee	Dillard University	8.23
12	37	Buck	Chappelow	bchappelow10@biglobe.ne.jp	1990-11-21	Student	Mohammad Ali Jinnah University	8.47
13	26	Marlee	Voas	mvoasp@cornell.edu	2000-12-16	Employee	Allahabad University	8.57

Total rows: 13 of 13 Query complete 00:00:00.056 Successfully run. Total query runtime: 56 msec. 13 rows affected.

➤ Output on Webpage

The screenshot shows a web-based database interface with a navigation bar:

- Hackathon Management System
- Home
- Participant Database
- Query Output** (highlighted)
- Hackathon Database

The results table has the same columns as the PostgreSQL output:

	part_id	first_name	last_name	email_add	dob	student	college	cpi
1	44	Darrel	Student	dStudent17@tinypic.com	Dec. 12, 1995	Student	Northwood University, Florida Campus	7.35
2	38	Mirabel	Bunton	mbunton11@stanford.edu	Jan. 3, 1992	Employee	Boston University	7.44
3	18	Kally	Radnedge	kradnedgeh@engadget.com	June 12, 1991	Employee	Chongqing Wenli University	7.46
4	39	Rosalinda	Dinjes	rdinjes12@ehow.com	April 19, 1991	Employee	Universidad de Zaragoza	7.47
5	11	Dorene	Disbury	ddisbursya@geocities.com	Sept. 5, 1999	Employee	Latvian Academy of Music	7.54
6	22	Sabra	Symms	ssy whole ms@360.cn	June 8, 1998	Employee	Lenoir-Rhyne College	7.62
7	6	Idette	Harback	iharback5@seattletimes.com	Oct. 15, 2000	Employee	Shanghai Institute of Foreign Trade	7.78
8	17	Jozef	Matis	jmati sg@bbb.org	Feb. 4, 1986	Student	Fairfield University	7.99
9	15	Emilio	Eves	eevese@dagondesign.com	Oct. 28, 1988	Student	University of International Business...	8.01
10	29	Francis	Gery	fgerys@imdb.com	Jan. 1, 1988	Student	Universidad Tecnológica del Perú	8.08
11	20	Christye	Hairy es	chairy es@elegantthemes.co...	Sept. 28, 1998	Employee	Dillard University	8.23
12	37	Buck	Chappelow	bchappelow10@biglobe.ne.jp	Nov. 21, 1990	Student	Mohammad Ali Jinnah University	8.47
13	26	Marlee	Voas	mvoasp@cornell.edu	Dec. 16, 2000	Employee	Allahabad University	8.57

Number of Tuples ➔ 13

SQL Query-2.

```
select * from participant where first_name like 'S%' or last_name like '%s' order by part_id;
```

Snapshot-2.

➤ Output in PostgreSQL

Query Query History

```
33
34  select * from participant where first_name like 'S%' or last_name like '%s' order by part_id;
35
```

Data output Messages Notifications

part_id [PK] bigint	first_name character varying	last_name character varying	email_add character varying	dob date	student character varying	college character varying	cpi double precision
1	4	Silvia	Calvey	scalvey3@pbs.org	2002-12-30	Employee	Georg-Simon-Ohm-Fachhochsch... 9.39
2	7	Sabine	Butterley	sbutterley6@earthlink.net	1998-12-24	Employee	University of Northern British Col... 9.15
3	12	Stanford	Hampe	shampeb@tripod.com	1988-01-26	Student	Knox College 4.59
4	13	Coriss	Antonopoulos	cantonopoulosc@cbslocal.c...	2003-08-06	Employee	Rowan University 5.08
5	15	Emilio	Eves	eewe3@dagondesign.com	1988-10-28	Student	University of International Busine... 8.01
6	17	Jozef	Matis	jmatisg@bbb.org	1986-02-04	Student	Fairfield University 7.99
7	20	Christye	Haireyes	chaires3@elegantthemes.co...	1998-09-28	Employee	Dillard University 8.23
8	21	Charmain	Brewls	cbrewlisk@360.cn	1990-02-03	Employee	American University in Cairo 4.56
9	22	Sabra	Symms	ssyymmsl@360.cn	1998-06-08	Employee	Lenoir-Rhyne College 7.62
10	25	Heall	Myers	hmyerso@multiply.com	1989-01-08	Student	Galilee College 5.96
11	26	Marlee	Voas	mvoasp@cornell.edu	2000-12-16	Employee	Allahabad University 8.57
12	39	Rosalinda	Dinjes	rdinjes12@ehow.com	1991-04-19	Employee	Universidad de Zaragoza 7.47
13	40	Shoshana	Gallelli	sgallelli13@icq.com	2000-07-08	Employee	Universidad Tecnológica de Pan... 6.76
14	43	Franzen	Sallans	fsallans16@vinaora.com	1991-09-23	Student	Sacred Heart University 6.84
15	45	Hermie	Perkins	hperkins18@state.gov	2004-01-05	Student	Northern Arizona University 6.56

Total rows: 15 of 15 Query complete 00:00:01.809 Ln 35, Col 1

➤ Output on Webpage

Hackathon Management System Home Participant Database **Query Output** Hackathon Database

part_id	first_name	last_name	email_add	dob	student	college	cpi
4	Silvia	Calvey	scalvey3@pbs.org	Dec. 30, 2002	Employee	Georg-Simon-Ohm-Fachhochschule Nürnberg	9.39
7	Sabine	Butterley	sbutterley6@earthlink.net	Dec. 24, 1998	Employee	University of Northern British Columbia	9.15
12	Stanford	Hampe	shampeb@tripod.com	Jan. 26, 1988	Student	Knox College	4.59
13	Coriss	Antonopoulos	cantonopoulosc@cbslocal.com	Aug. 6, 2003	Employee	Rowan University	5.08
15	Emilio	Eves	eewe3@dagondesign.com	Oct. 28, 1988	Student	University of International Business	8.01
17	Jozef	Matis	jmatisg@bbb.org	Feb. 4, 1986	Student	Fairfield University	7.99
20	Christye	Haireyes	chaires3@elegantthemes.com	Sept. 28, 1998	Employee	Dillard University	8.23
21	Charmain	Brewls	cbrewlisk@360.cn	Feb. 3, 1990	Employee	American University in Cairo	4.56
22	Sabra	Symms	ssyymmsl@360.cn	June 8, 1998	Employee	Lenoir-Rhyne College	7.62
25	Heall	Myers	hmyerso@multiply.com	Jan. 8, 1989	Student	Galilee College	5.96
26	Marlee	Voas	mvoasp@cornell.edu	Dec. 16, 2000	Employee	Allahabad University	8.57
39	Rosalinda	Dinjes	rdinjes12@ehow.com	April 19, 1991	Employee	Universidad de Zaragoza	7.47
40	Shoshana	Gallelli	sgallelli13@icq.com	July 8, 2000	Employee	Universidad Tecnológica de Panamá	6.76
43	Franzen	Sallans	fsallans16@vinaora.com	Sept. 23, 1991	Student	Sacred Heart University	6.84
45	Hermie	Perkins	hperkins18@state.gov	Jan. 5, 2004	Student	Northern Arizona University	6.56

Number of Tuples ➔ 15