AMRITA
VISHWA VIDYAPEETHAM

Kalotsavam Database Management System

Group:

Bhanu Prakash	AM.EN.U4CSE17053
Harikrishna	AM.EN.U4CSE17024
Manikanta	AM.EN.U4CSE17012
Sumanth	AM.EN.U4CSE17019

Under the Supervision and Guidance of

Mrs. Asha Ashok

Asst. Professor

And

Mrs. Prathibhamol C. P.

Asst. Professor

Kalotsavam Database Management System

Abstract

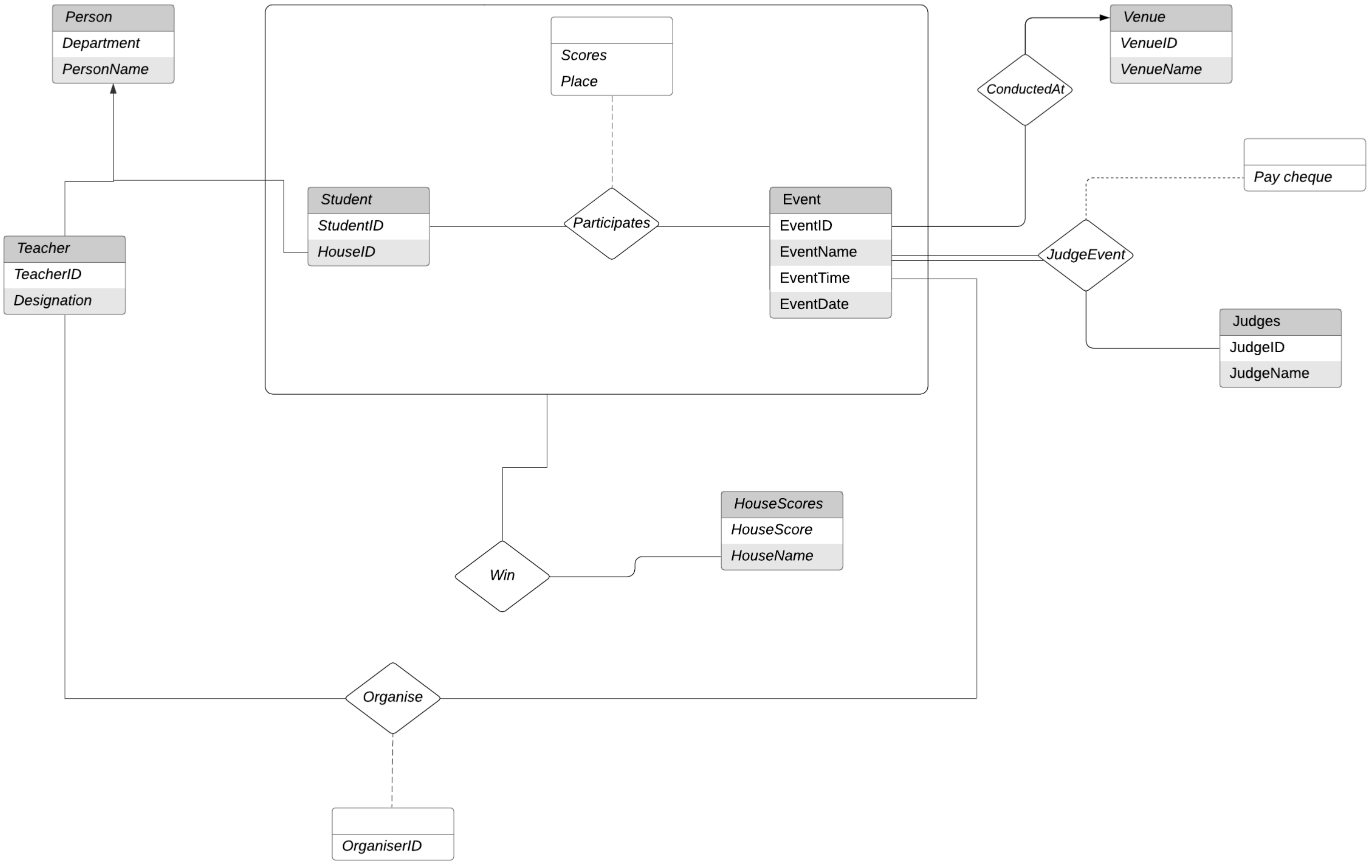
The university conducts Amrita Kalotsavam every year. The system will be storing the details of events, schedules, participation, scores, judges, venues, winners. This database system will also keep track of the coordinators involved, the venue details and its availability required during Kalotsavam. There will be different categories of requirements mentioned for each event conducted and the Finance related to it. Participant feedback will also be collected regarding all the events. There will be different entities like Staff, Co-ordinators, Expenditure, Events, Service Providers, and a lot of other required entities. The whole database can be divided into three parts: 1. Events, 2. Show, Performers and Equipment and 3. Co-ordinators and Staff.

The ‘Events’ subject area is the central part of our model. In this part we’ll store the most important details about our events. As mentioned in the introduction, each event can have several sub-events. A show is a single sub-event, focused on one topic, having at least one performer, etc. Will keep a record of any show associated with past, present, and future events. The subject area of the third part of the model is about co-ordinators and their roles.

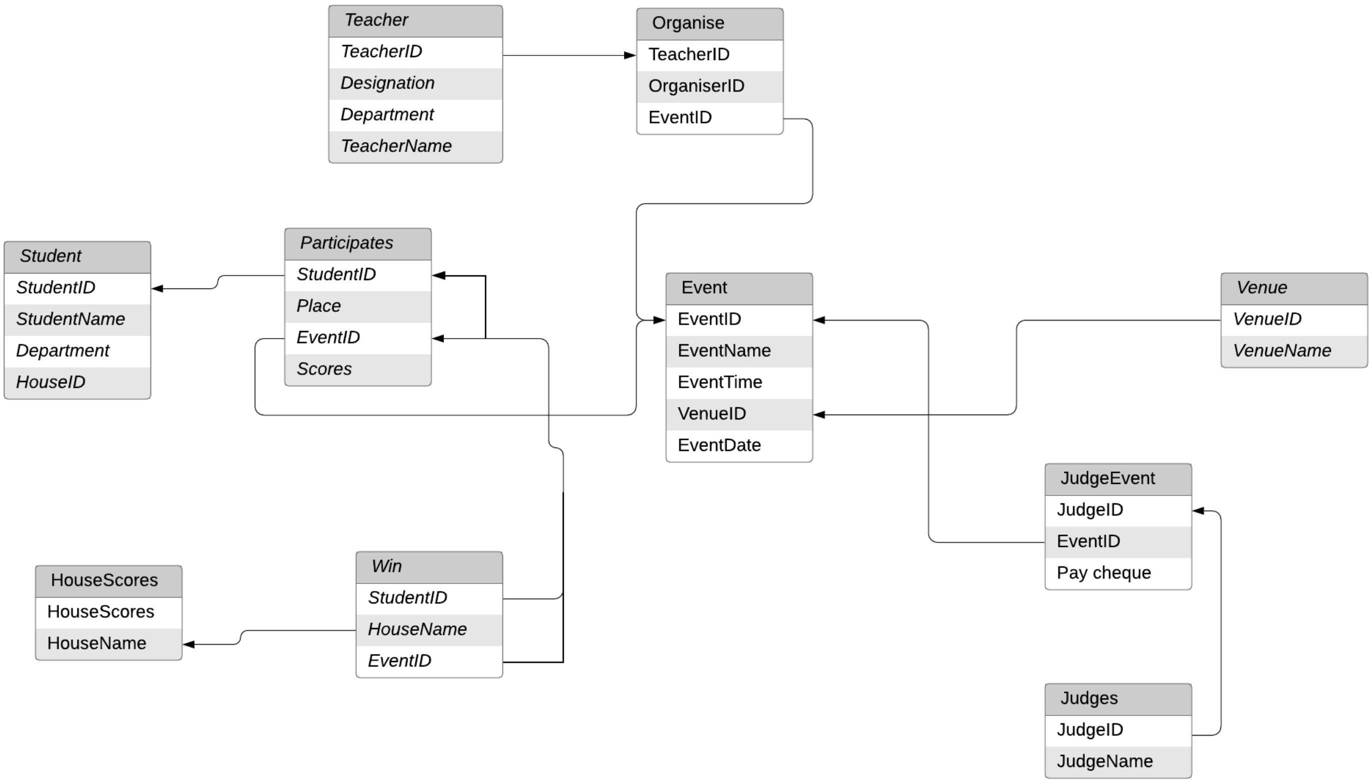
Entity Relationship Diagram

DBMS ER Diagram (UML Notation)

Bhanu, Harikrishna, Manikanta, Sumanth | November 12, 2019



Schema Diagram



Kalotsavam Database Management System

Kalotsavam(StudentID, StudentName, Department, HouseID, Place, **EventID**, EventName, EventTime, EventDate, Scores, **TeacherID**, OrganiserID, Designation, TeacherName, VenueID, VenueName, **JudgeID**, JudgeName, PayCheque,HouseName, HouseScores)

Primary Key: (StudentID, EventID, TeacherID, JudgeID)

The above mentioned table consists of all the attributes required for the database.

Assumptions/Constraints

StudentID:	ID assigned to each student
StudentName:	name of the student
Department:	name of the department the teacher or student belongs to
HouseID:	ID assigned to each House
Place:	the position secured by each student in an event
EventID:	ID assigned to each occurring event
EventName:	name of each event
EventTime:	scheduled time of the event
EventDate:	date of the event conducted
Scores:	score attained by the student for an event
TeacherID:	ID assigned to a particular teacher
OrganiserID:	ID assigned to a teacher who is an organizer
Designation:	designation of the teacher
TeacherName:	name of each teacher
VenueID:	ID assigned to each venue
VenueName:	name of each venue
JudgeID:	ID assigned to each judge
JudgeName:	name of each judge
HouseName:	name of each house
HouseScores:	scores attained by each respective house
PayCheque:	amount given to each judge for an event

Normalization

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

Table in 1NF

A relation is in **first normal form** if and only if the domain of each attribute contains only atomic values, and the value of each attribute contains only a single value from that domain.

There are no multi valued attributes and all the attributes are atomic, so there will be no change in the relational schema.

Kalotsavam(<u>StudentID</u>, StudentName, Department, HouseID, Place, **<u>EventID</u>**, EventName, EventTime, EventDate, Scores, **<u>TeacherID</u>**, OrganiserID, Designation, TeacherName, VenueID, VenueName, **<u>JudgeID</u>**, JudgeName, Paycheque, HouseName, HouseScores)

Functional Dependencies

StudentID →StudentName, Department, HouseID
EventID →EventName, EventTIme, EventDate, VenueID, VenueName
JudgeID →JudgeName
TeacherID →TeacherName, Designation, Department
StudentID, EventID →Place, Scores, HouseName, HouseScores
EventID, TeacherID →OrganiserID
JudgeID, EventID →PayCheque

- In 2NF, relational must be in 1NF.
- A functional dependency on the part of any candidate key is a violation of 2NF. In addition to the primary key, the relation may contain other candidate keys; it is necessary to establish that no non-prime attributes have partial dependencies on *any* of these candidate keys

Tables in 2NF

Student(StudentID, StudentName, Department, HouseID)
Event(EventID,EventName, EventTIme, EventDate, VenueID, VenueName)
Judge(JudgeID, JudgeName)
Teacher(TeacherID, TeacherName, Designation, Department)
Participates(StudentID, EventID, Place, Scores, HouseName, HouseScores)
Organiser(EventID, TeacherID, OrganiserID)
JudgeEvent(JudgeID, EventID, PayCheque)

According to the functional dependencies the new tables are formed in the 2NF form.

Transitive Dependencies

EventID →VenueID →VenueName

- A relation is in 3NF if it is in 2NF and not contain any transitive partial dependency.
- 3NF is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

Tables in 3NF

Student(StudentID, StudentName, Department, HouseID)
Venue(VenueID, VenueName)
Event(EventID,EventName, EventTIme, EventDate, VenueID)
Judge(JudgeID, JudgeName)
Teacher(TeacherID, TeacherName, Designation, Department)
Participates(StudentID, EventID, Place, Scores, HouseName, HouseScores)
Organiser(EventID, TeacherID, OrganiserID)

JudgeEvent(JudgeID, EventID, PayCheque)

Comparison of ER Design model after normalization procedure

The initial ER Design had a total of 10 tables. After normalization the number of tables reduced to 8 tables. There was data redundancy in the initial model which was reduced in the normalized form.

Win and ***HouseScores*** tables has been removed and instead the attributes - HouseName and HouseScores are put into ***Participates*** table reducing redundancy.

Now the database design has become more flexible than before.

DDL Statements

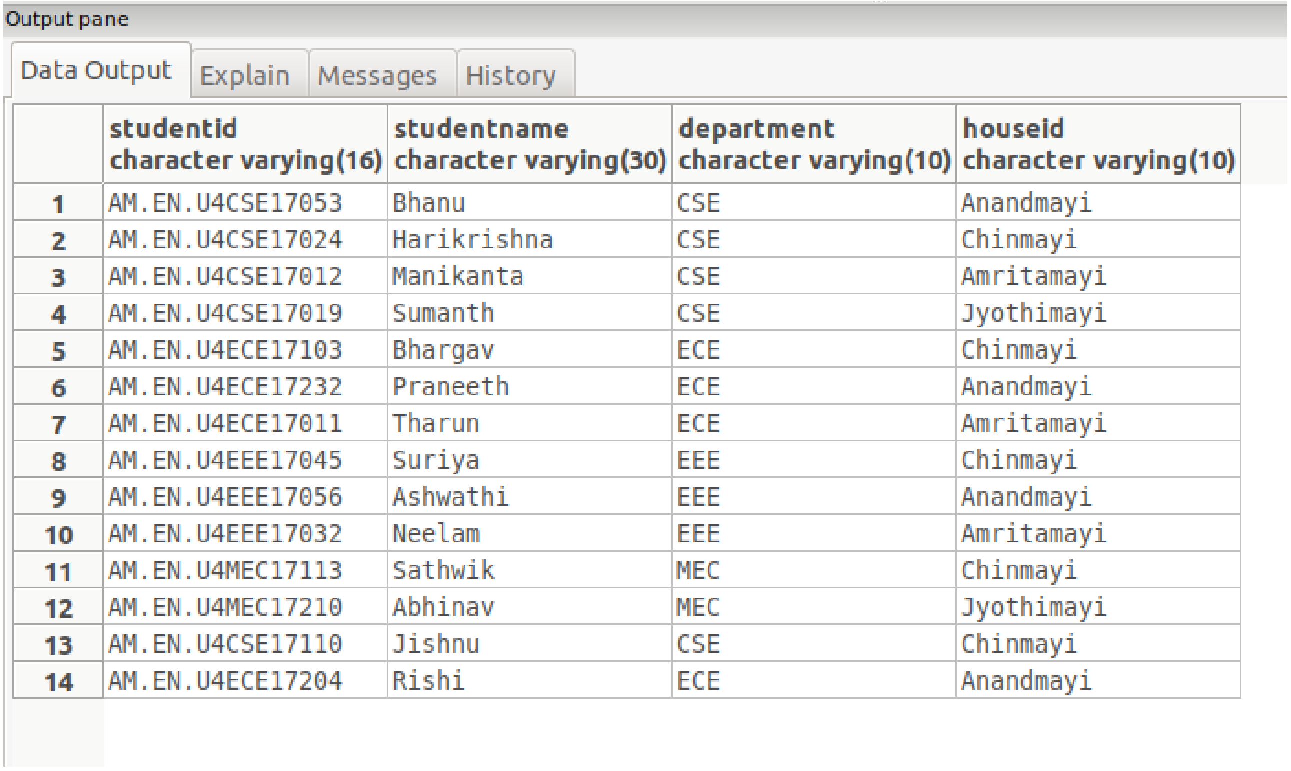
```
CREATE TABLE Student(
StudentID varchar(16),
StudentName varchar(30),
Department varchar(10),
HouseID varchar(10),
Primary key(StudentID)
);

CREATE TABLE Venue(
VenueID varchar(10),
VenueName varchar(20),
Primary key(VenueID)
);

CREATE TABLE Event(
EventID varchar(15),
EventName varchar(20),
EventTIme varchar(10),
EventDate date,
VenueID varchar(10),
Primary key(EventID),
Foreign key(VenueID) references Venue,
);

CREATE TABLE Judge(
JudgeID varchar(10),
JudgeName varchar(20),
Primary key(JudgeID)
);

CREATE TABLE Teacher(
TeacherID varchar(10),
TeacherName varchar(20 ),
Designation varchar(10),
Department varchar(10),
Primary key(TeacherID)
);

CREATE TABLE Participates(
 StudentID varchar(16),
 EventID varchar(15),
 Place int(3),
 Scores int,
 Foreign key(StudentID) references Student,
 Foreign key(EventID) references Event,
 Primary key(StudentID, EventID)
);

CREATE TABLE Organiser(
 EventID varchar(15),
 TeacherID varchar(10),
 OrganiserID varchar(10),
 Foreign key(EventID) references Event,
 Foreign key(TeacherID) references Teacher,
 Primary key(EventID, TeacherID)
);

CREATE TABLE JudgeEvent(
 JudgeID varchar(10),
 EventID varchar(15),
 PayCheque int,
 Foreign key(JudgeID) references Judge,
 Foreign key(EventID) references Event,
 Primary key(JudgeID, EventID)
);
```

DML STATEMENTS

```
INSERT INTO Student VALUES
('AM.EN.U4CSE17053', 'Bhanu', 'CSE', 'Anandmayi'),
('AM.EN.U4CSE17024', 'Harikrishna', 'CSE', 'Chinmayi'),
('AM.EN.U4CSE17012', 'Manikanta', 'CSE', 'Amritamayi'),
('AM.EN.U4CSE17019', 'Sumanth', 'CSE', 'Jyothimayi'),
('AM.EN.U4ECE17103', 'Bhargav', 'ECE', 'Chinmayi'),
('AM.EN.U4ECE17232', 'Praneeth', 'ECE', 'Anandmayi'),
('AM.EN.U4ECE17011', 'Tharun', 'ECE', 'Amritamayi'),
('AM.EN.U4EEE17045', 'Suriya', 'EEE', 'Chinmayi'),
('AM.EN.U4EEE17056', 'Ashwathi', 'EEE','Anandmayi'),
('AM.EN.U4EEE17032', 'Neelam', 'EEE', 'Amritamayi'),
('AM.EN.U4MEC17113', 'Sathwik', 'MEC', 'Chinmayi'),
('AM.EN.U4MEC17210', 'Abhinav', 'MEC', 'Jyothimayi');
('AM.EN.U4CSE17110', 'Jishnu', 'CSE', 'Chinmayi'),
('AM.EN.U4ECE17204', 'Rishi', 'ECE', 'Anandmayi')

INSERT INTO Venue VALUES
('V01', 'Amriteshwari Hall'),
('V02', 'Acharya Hall'),
('V03', 'Shraddha Hall'),
('V04', 'Open Stage'),
('V05', 'Ground');


INSERT INTO Event VALUES
('E01', 'Dancing Divas', '17:00', 'V01', '2019-04-02'),
('E02', 'Nightingales', '10:00', 'V03', '2019-04-02'),
('E03', 'Fashion Night', '13:00', 'V01', '2019-04-02'),
('E04', 'Face Painting', '16:00', 'V04', '2019-04-03'),
('E05', 'Gaming', '18:00', 'V05', '2019-04-03');


INSERT INTO Judge VALUES
('J01','Mrs.Geetha'),
('J02', 'Mr. Swaminathan'),
('J03', 'Mr.Sarath'),
('J04', 'Mrs.Jyothisha'),
('J05', 'Mr. Sajeev');

INSERT INTO Teacher VALUES
('T01','Krishna','Asst. Prof','CSE'),
('T02','Sarath','Asst. Prof','CSE'),
('T03','Asha Ashok','Asst. Prof','CSE'),
('T04','Pratibha','Prof','CSE'),
('T05','Parvathi','Prof','EEE'),
('T06','Thara','Prof','ECE'),
('T07','Arun','Asst. Prof','MEC')


INSERT INTO Participates VALUES
('AM.EN.U4CSE17053', 'E01', '1', 100),
('AM.EN.U4CSE17024', 'E03', '2', 45),
('AM.EN.U4CSE17012', 'E01', '2', 75),
('AM.EN.U4CSE17019', 'E05', '2', 60),
('AM.EN.U4ECE17103', 'E03', '1', 60),
('AM.EN.U4ECE17232', 'E02', '0', 60),
('AM.EN.U4ECE17011', 'E02', '3', 77),
('AM.EN.U4EEE17045', 'E04', '0', 34),
('AM.EN.U4EEE17056', 'E05', '0', 20),
('AM.EN.U4EEE17032', 'E04', '0', 36),
('AM.EN.U4MEC17113', 'E02', '1', 105),
('AM.EN.U4MEC17210', 'E05', '0', 20),
('AM.EN.U4ECE17011', 'E03', '0', 15),
('AM.EN.U4CSE17024', 'E01', '3', 50),
('AM.EN.U4CSE17019', 'E01', '0', 10),
('AM.EN.U4ECE17103', 'E01', '0', 30),
('AM.EN.U4ECE17232', 'E03', '3', 30),
('AM.EN.U4MEC17113', 'E03', '0', 10),
('AM.EN.U4MEC17210', 'E02', '2', 86),
('AM.EN.U4CSE17053', 'E02', '0', 27),
('AM.EN.U4EEE17045', 'E05', '1', 80),
('AM.EN.U4ECE17011', 'E05', '3', 50),
('AM.EN.U4MEC17113', 'E04', '1', 98),
('AM.EN.U4EEE17056', 'E04', '2', 79),
('AM.EN.U4MEC17210', 'E04', '3', 56)


INSERT INTO Organiser VALUES
('E01','T01','OG1'),
('E01','T03','OG2'),
('E01','T05','OG3'),
('E02','T01','OG1'),
('E02','T02','OG4'),
('E02','T03','OG2'),
('E03','T03','OG2'),
('E03','T02','OG5'),
('E03','T05','OG3'),
('E04','T01','OG1'),
('E04','T05','OG3'),
('E04','T02','OG4'),
('E05','T03','OG2'),
('E05','T01','OG1'),
('E05','T04','OG5')

INSERT INTO JudgeEvent VALUES
('J01','E01',10000),
('J03','E01',10000),
('J04','E01',10000),
('J02','E02',20000),
('J05','E02',20000),
('J01','E02',20000),
('J04','E03',10000),
('J02','E03',10000),
('J05','E03',10000),
('J03','E04',15000),
('J02','E04',15000),
('J01','E04',15000),
('J04','E05',20000),
('J05','E05',20000),
('J02','E05',20000)
```

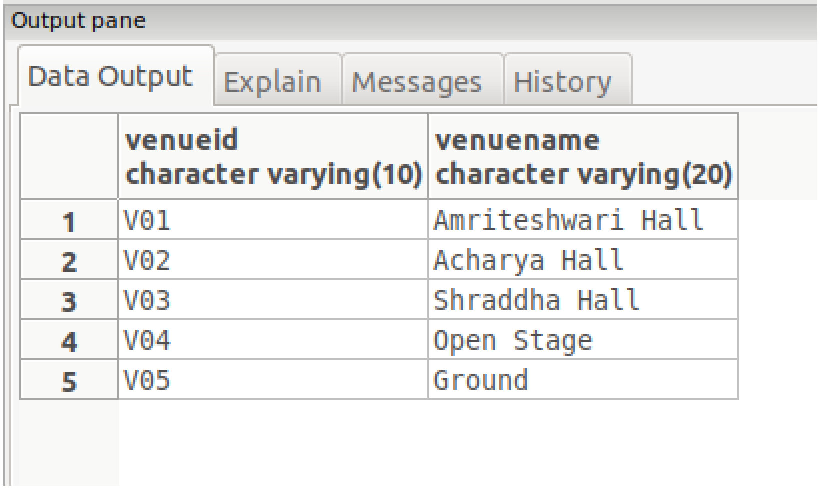
SELECT STATEMENTS

select * from Student;



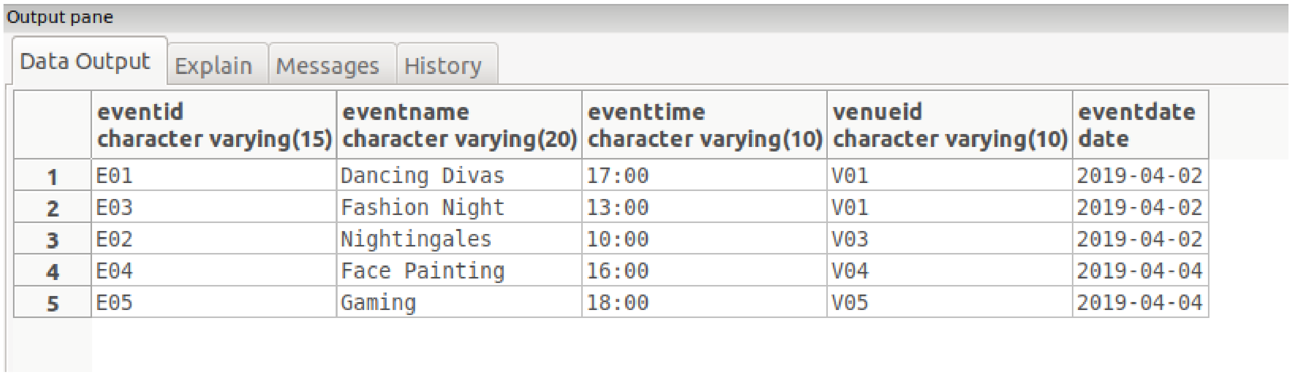
	studentid character varying(16)	studentname character varying(30)	department character varying(10)	houseid character varying(10)
1	AM.EN.U4CSE17053	Bhanu	CSE	Anandmayi
2	AM.EN.U4CSE17024	Harikrishna	CSE	Chinmayi
3	AM.EN.U4CSE17012	Manikanta	CSE	Amritamayi
4	AM.EN.U4CSE17019	Sumanth	CSE	Jyothimayi
5	AM.EN.U4ECE17103	Bhargav	ECE	Chinmayi
6	AM.EN.U4ECE17232	Praneeth	ECE	Anandmayi
7	AM.EN.U4ECE17011	Tharun	ECE	Amritamayi
8	AM.EN.U4EEE17045	Suriya	EEE	Chinmayi
9	AM.EN.U4EEE17056	Ashwathi	EEE	Anandmayi
10	AM.EN.U4EEE17032	Neelam	EEE	Amritamayi
11	AM.EN.U4MEC17113	Sathwik	MEC	Chinmayi
12	AM.EN.U4MEC17210	Abhinav	MEC	Jyothimayi
13	AM.EN.U4CSE17110	Jishnu	CSE	Chinmayi
14	AM.EN.U4ECE17204	Rishi	ECE	Anandmayi

select * from Venue;



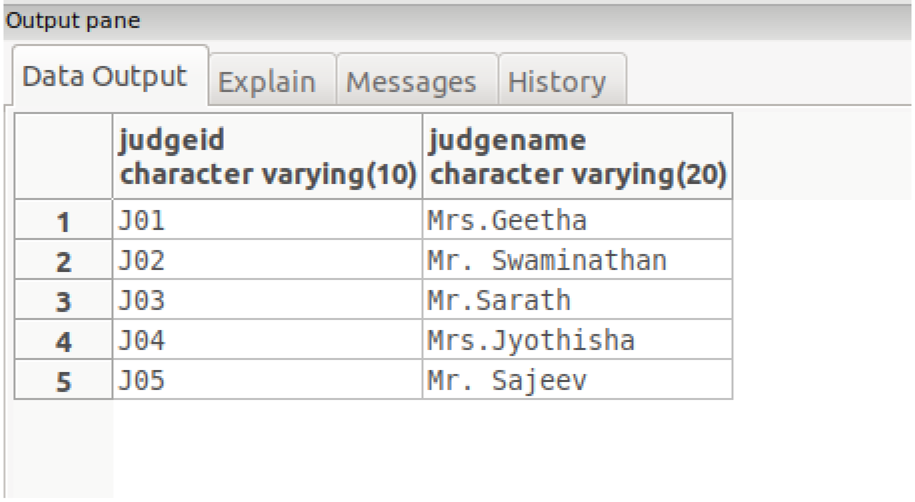
	venueid character varying(10)	venuename character varying(20)
1	V01	Amriteshwari Hall
2	V02	Acharya Hall
3	V03	Shraddha Hall
4	V04	Open Stage
5	V05	Ground

select * from Event;



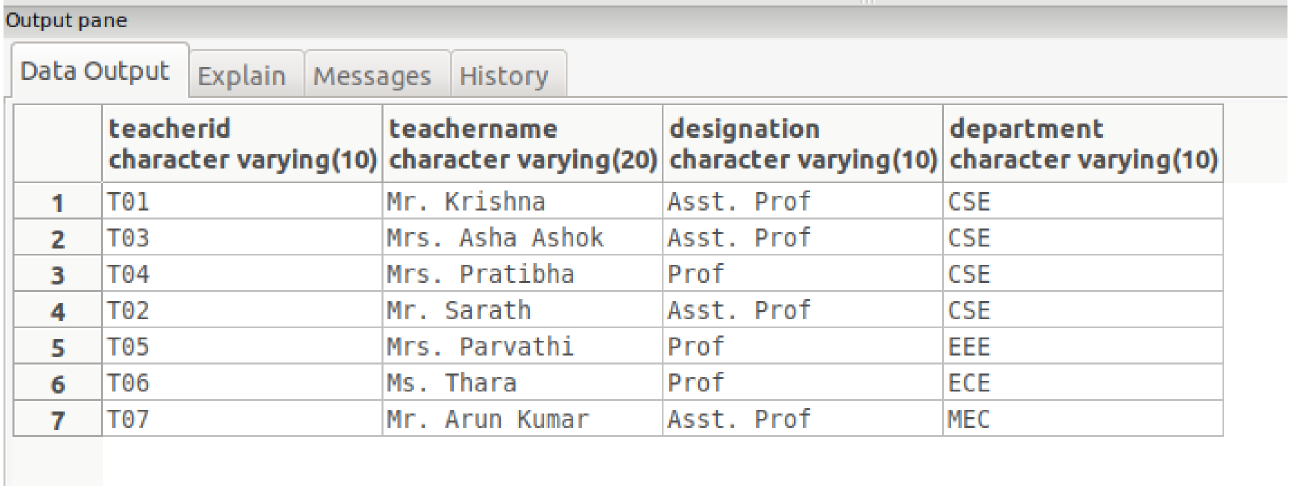
	eventid character varying(15)	eventname character varying(20)	eventtime character varying(10)	venueid character varying(10)	eventdate date
1	E01	Dancing Divas	17:00	V01	2019-04-02
2	E03	Fashion Night	13:00	V01	2019-04-02
3	E02	Nightingales	10:00	V03	2019-04-02
4	E04	Face Painting	16:00	V04	2019-04-04
5	E05	Gaming	18:00	V05	2019-04-04

select * from Judge;



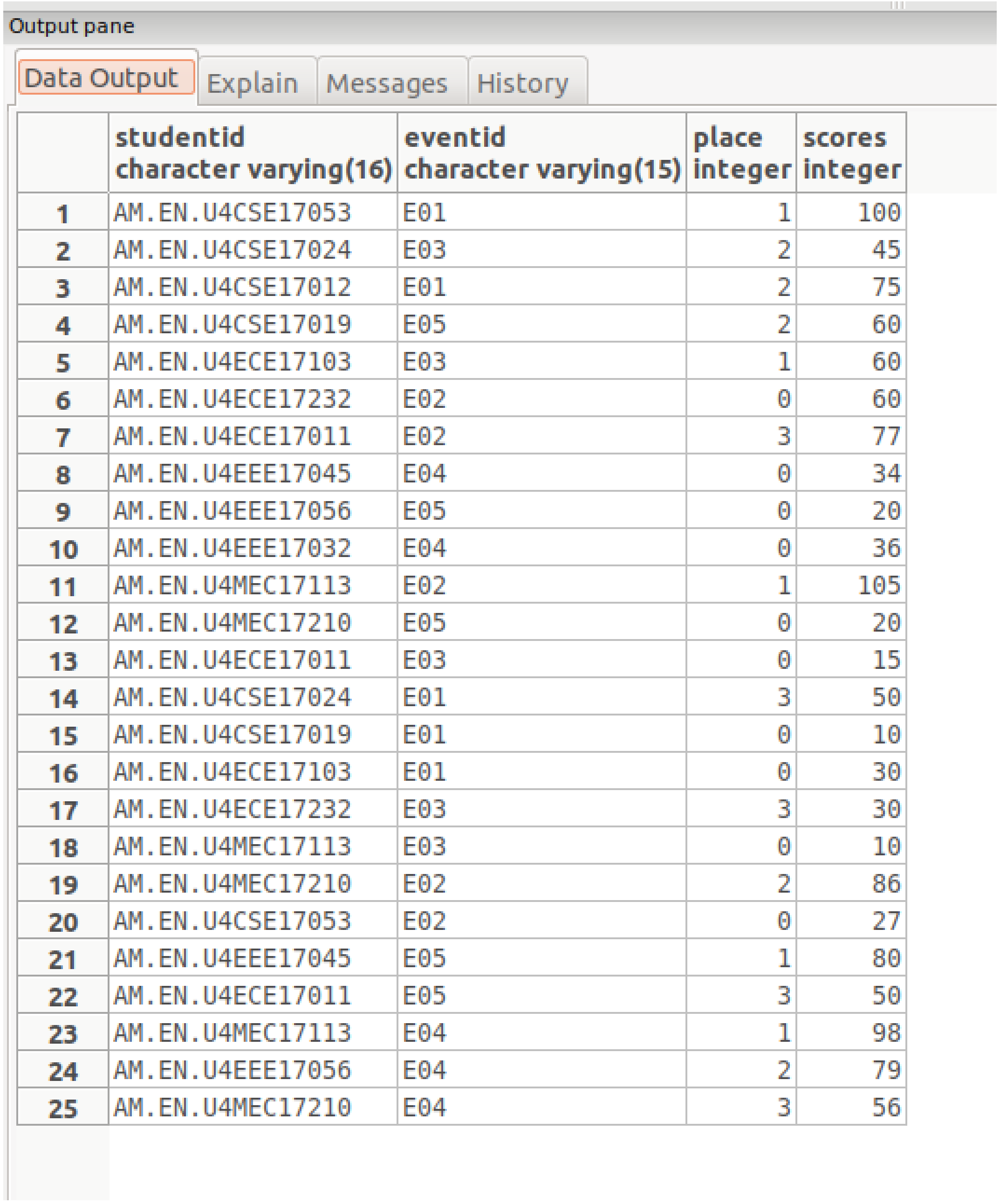
	judgeid character varying(10)	judgename character varying(20)
1	J01	Mrs.Geetha
2	J02	Mr. Swaminathan
3	J03	Mr.Sarath
4	J04	Mrs.Jyothisha
5	J05	Mr. Sajeev

select * from Teacher;



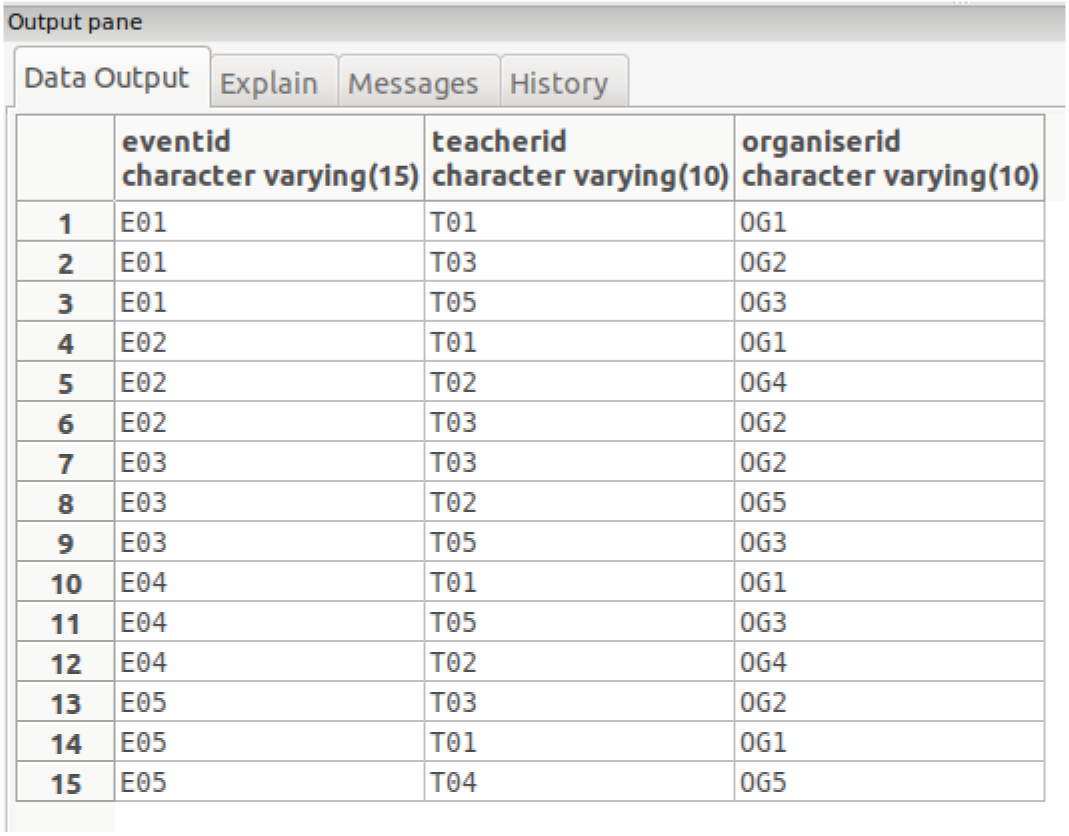
	teacherid character varying(10)	teachername character varying(20)	designation character varying(10)	department character varying(10)
1	T01	Mr. Krishna	Asst. Prof	CSE
2	T03	Mrs. Asha Ashok	Asst. Prof	CSE
3	T04	Mrs. Pratibha	Prof	CSE
4	T02	Mr. Sarath	Asst. Prof	CSE
5	T05	Mrs. Parvathi	Prof	EEE
6	T06	Ms. Thara	Prof	ECE
7	T07	Mr. Arun Kumar	Asst. Prof	MEC

select * from Participates;



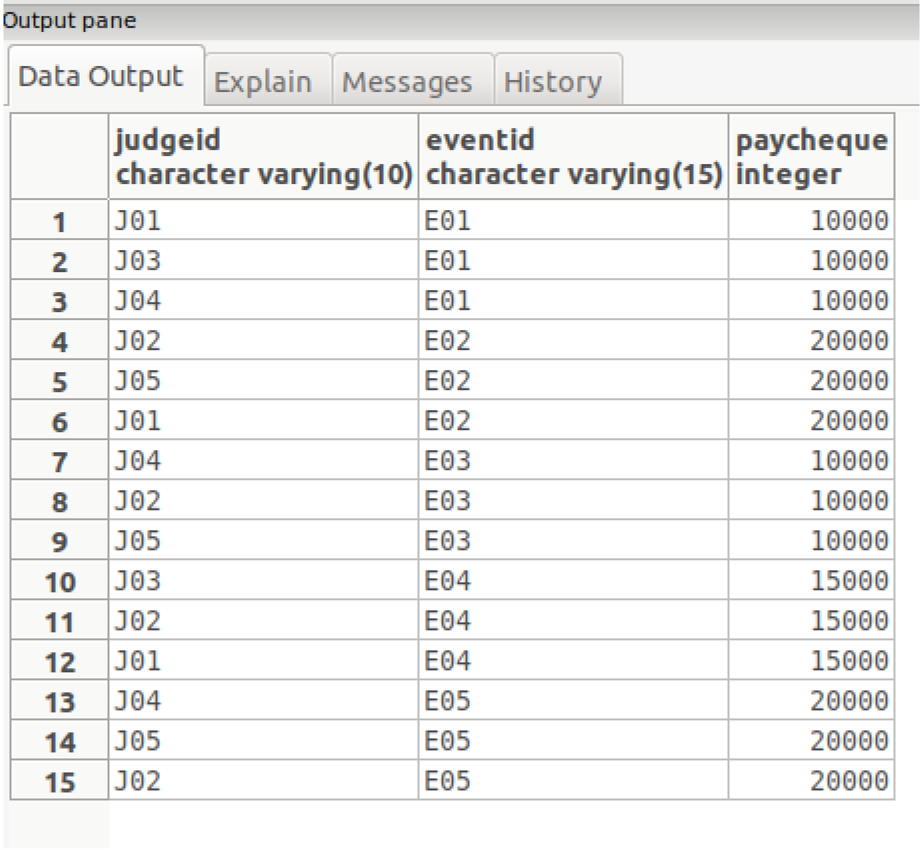
	studentid character varying(16)	eventid character varying(15)	place integer	scores integer
1	AM.EN.U4CSE17053	E01	1	100
2	AM.EN.U4CSE17024	E03	2	45
3	AM.EN.U4CSE17012	E01	2	75
4	AM.EN.U4CSE17019	E05	2	60
5	AM.EN.U4ECE17103	E03	1	60
6	AM.EN.U4ECE17232	E02	0	60
7	AM.EN.U4ECE17011	E02	3	77
8	AM.EN.U4EEE17045	E04	0	34
9	AM.EN.U4EEE17056	E05	0	20
10	AM.EN.U4EEE17032	E04	0	36
11	AM.EN.U4MEC17113	E02	1	105
12	AM.EN.U4MEC17210	E05	0	20
13	AM.EN.U4ECE17011	E03	0	15
14	AM.EN.U4CSE17024	E01	3	50
15	AM.EN.U4CSE17019	E01	0	10
16	AM.EN.U4ECE17103	E01	0	30
17	AM.EN.U4ECE17232	E03	3	30
18	AM.EN.U4MEC17113	E03	0	10
19	AM.EN.U4MEC17210	E02	2	86
20	AM.EN.U4CSE17053	E02	0	27
21	AM.EN.U4EEE17045	E05	1	80
22	AM.EN.U4ECE17011	E05	3	50
23	AM.EN.U4MEC17113	E04	1	98
24	AM.EN.U4EEE17056	E04	2	79
25	AM.EN.U4MEC17210	E04	3	56

select * from Organiser;



	eventid character varying(15)	teacherid character varying(10)	organiserid character varying(10)
1	E01	T01	OG1
2	E01	T03	OG2
3	E01	T05	OG3
4	E02	T01	OG1
5	E02	T02	OG4
6	E02	T03	OG2
7	E03	T03	OG2
8	E03	T02	OG5
9	E03	T05	OG3
10	E04	T01	OG1
11	E04	T05	OG3
12	E04	T02	OG4
13	E05	T03	OG2
14	E05	T01	OG1
15	E05	T04	OG5

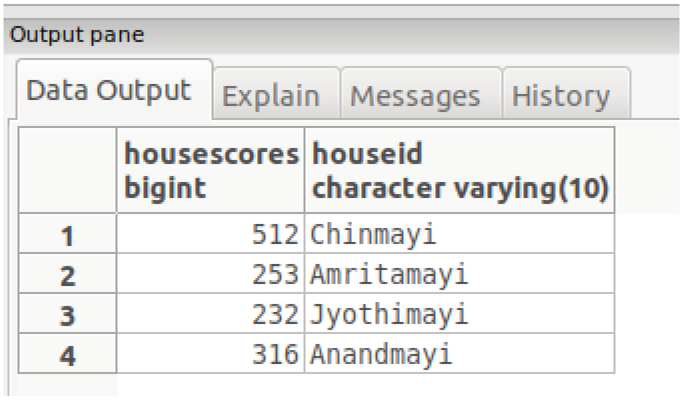
Select * from JudgeEvent



	judgeid character varying(10)	eventid character varying(15)	paycheque integer
1	J01	E01	10000
2	J03	E01	10000
3	J04	E01	10000
4	J02	E02	20000
5	J05	E02	20000
6	J01	E02	20000
7	J04	E03	10000
8	J02	E03	10000
9	J05	E03	10000
10	J03	E04	15000
11	J02	E04	15000
12	J01	E04	15000
13	J04	E05	20000
14	J05	E05	20000
15	J02	E05	20000

1. Aggregate Functions

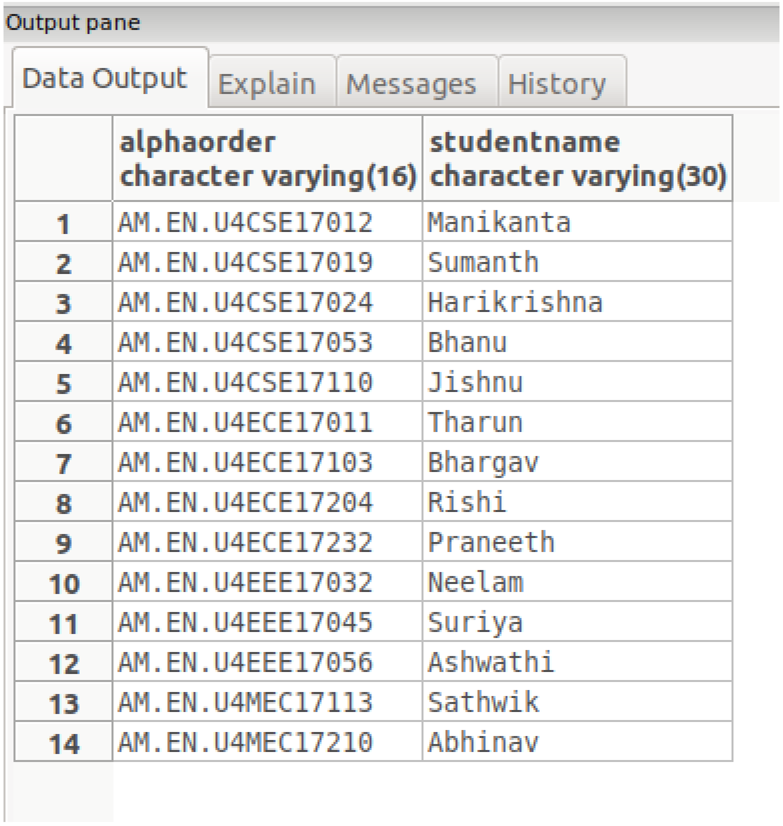
select sum(Scores) as HouseScores, HouseID from Participates natural join Student group by HouseID;



	housescores bigint	houseid character varying(10)
1	512	Chinmayi
2	253	Amritamayi
3	232	Jyothimayi
4	316	Anandmayi

2. Order by

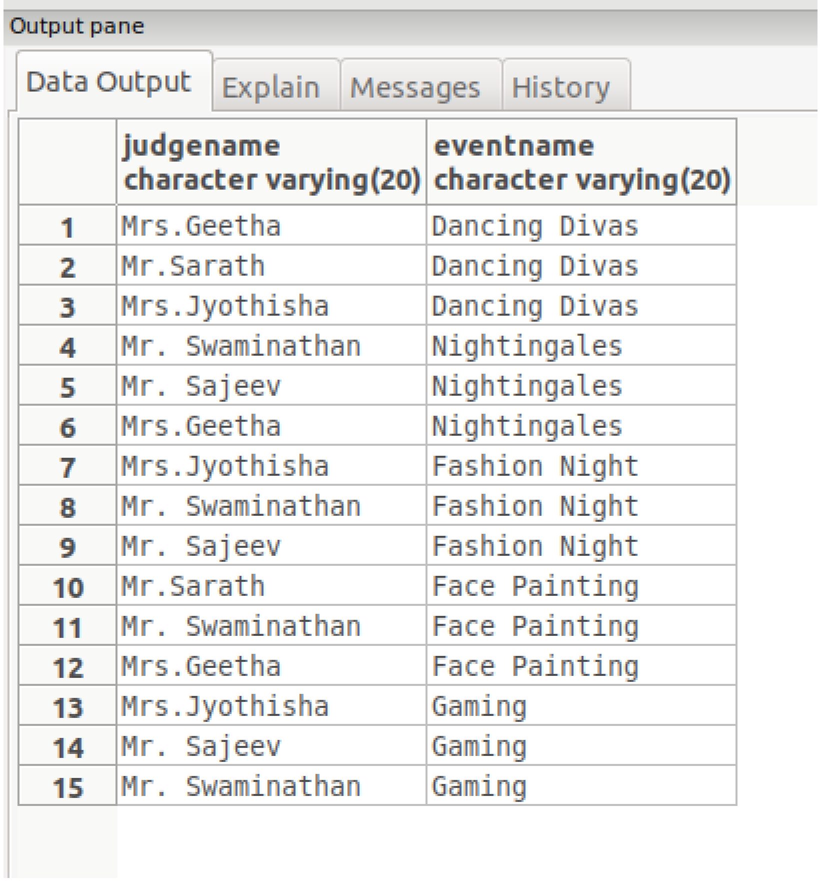
select StudentID as AlphaOrder, StudentName from Student order by StudentID



	alphaorder character varying(16)	studentname character varying(30)
1	AM.EN.U4CSE17012	Manikanta
2	AM.EN.U4CSE17019	Sumanth
3	AM.EN.U4CSE17024	Harikrishna
4	AM.EN.U4CSE17053	Bhanu
5	AM.EN.U4CSE17110	Jishnu
6	AM.EN.U4ECE17011	Tharun
7	AM.EN.U4ECE17103	Bhargav
8	AM.EN.U4ECE17204	Rishi
9	AM.EN.U4ECE17232	Praneeth
10	AM.EN.U4EEE17032	Neelam
11	AM.EN.U4EEE17045	Suriya
12	AM.EN.U4EEE17056	Ashwathi
13	AM.EN.U4MEC17113	Sathwik
14	AM.EN.U4MEC17210	Abhinav

3. Join, Outer Join

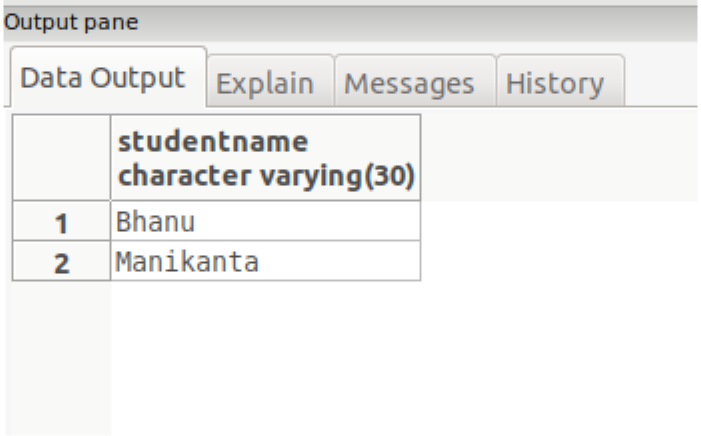
select JudgeName,EventName from (Judge join JudgeEvent on Judge.JudgeID = JudgeEvent.JudgeID) as A join Event on A.EventID=Event.EventID;



	judgename	eventname
	character varying(20)	character varying(20)
1	Mrs.Geetha	Dancing Divas
2	Mr.Sarath	Dancing Divas
3	Mrs.Jyothisha	Dancing Divas
4	Mr. Swaminathan	Nightingales
5	Mr. Sajeev	Nightingales
6	Mrs.Geetha	Nightingales
7	Mrs.Jyothisha	Fashion Night
8	Mr. Swaminathan	Fashion Night
9	Mr. Sajeev	Fashion Night
10	Mr.Sarath	Face Painting
11	Mr. Swaminathan	Face Painting
12	Mrs.Geetha	Face Painting
13	Mrs.Jyothisha	Gaming
14	Mr. Sajeev	Gaming
15	Mr. Swaminathan	Gaming

4. Query having Boolean Operators

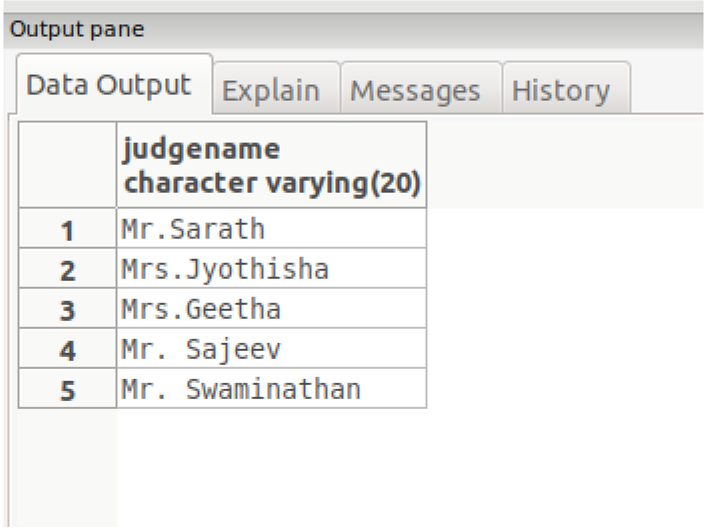
select StudentName from Student natural join Participates natural join Event where EventName = 'Dancing Divas' and Scores>50



	studentname
	character varying(30)
1	Bhanu
2	Manikanta

5. Query having Arithmetic Operators

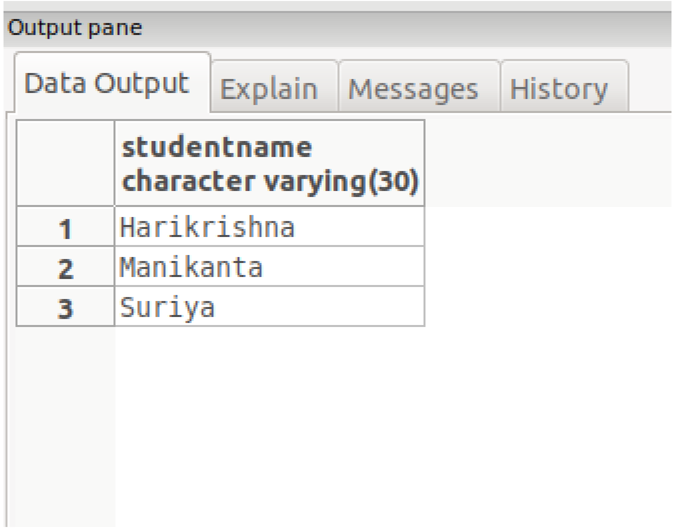
select distinct JudgeName from JudgeEvent natural join Judge where PayCheque > 10000;



	judgename character varying(20)
1	Mr.Sarath
2	Mrs.Jyothisha
3	Mrs.Geetha
4	Mr. Sajeev
5	Mr. Swaminathan

6. Search query using String operators

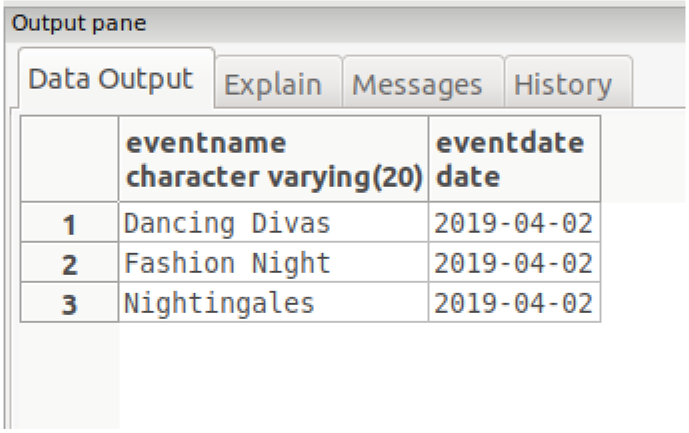
select StudentName from Student where StudentName like '%a'



	studentname character varying(30)
1	Harikrishna
2	Manikanta
3	Suriya

7. Use of to_char, extract

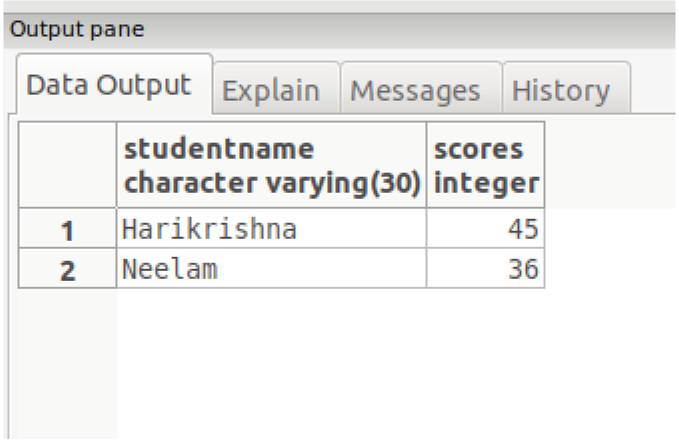
select EventName, EventDate from Event where extract(day from EventDate) = '02';



	eventname character varying(20)	eventdate date
1	Dancing Divas	2019-04-02
2	Fashion Night	2019-04-02
3	Nightingales	2019-04-02

8. Between, IN, Not Between, Not IN

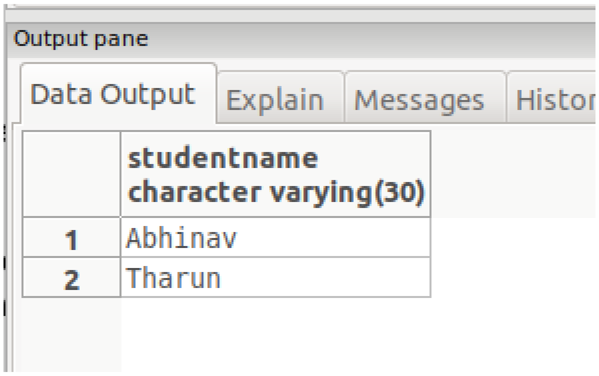
select distinct StudentName, Scores from Student natural join Participates where Scores between 35 and 49;



	studentname character varying(30)	scores integer
1	Harikrishna	45
2	Neelam	36

9. Set Operations

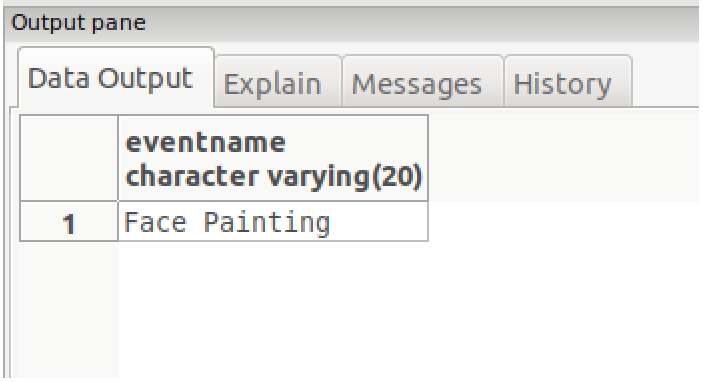
select Studentname from Student natural join Participates where EventID = 'E02' intersect select Studentname from Student natural join Participates where EventID = 'E05';



	studentname character varying(30)
1	Abhinav
2	Tharun

10. Subquery

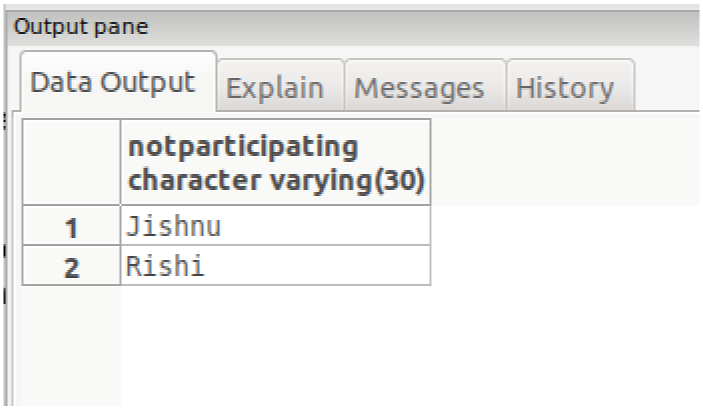
select EventName from Event where VenueID in (select VenueID from Venue where VenueName='Open Stage');



	eventname character varying(20)
1	Face Painting

11. Subquery using Exists

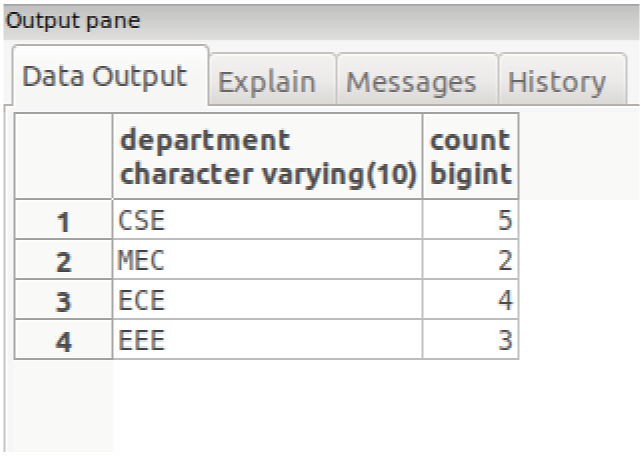
select StudentName as NOTParticipating from Student where not exists(select * from (Student natural join Participates) as A where Student.StudentID = A.StudentID)



	notparticipating character varying(30)
1	Jishnu
2	Rishi

No. of Students in each Department

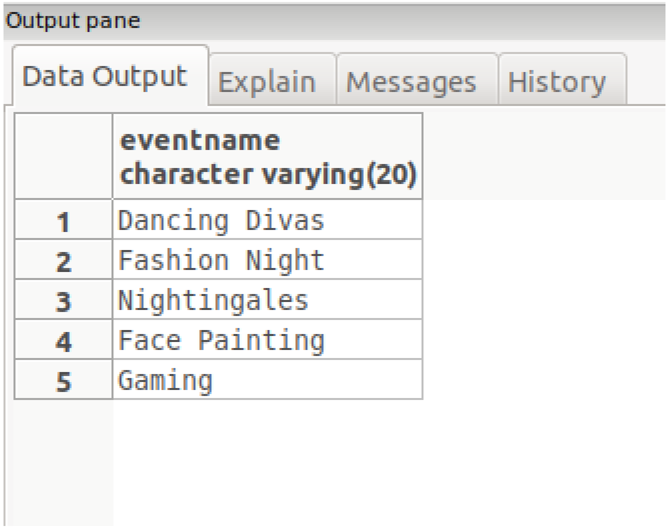
select Department, count(StudentID) from Student group by Department having count(StudentID) > 0;



	department character varying(10)	count bigint
1	CSE	5
2	MEC	2
3	ECE	4
4	EEE	3

List of Events

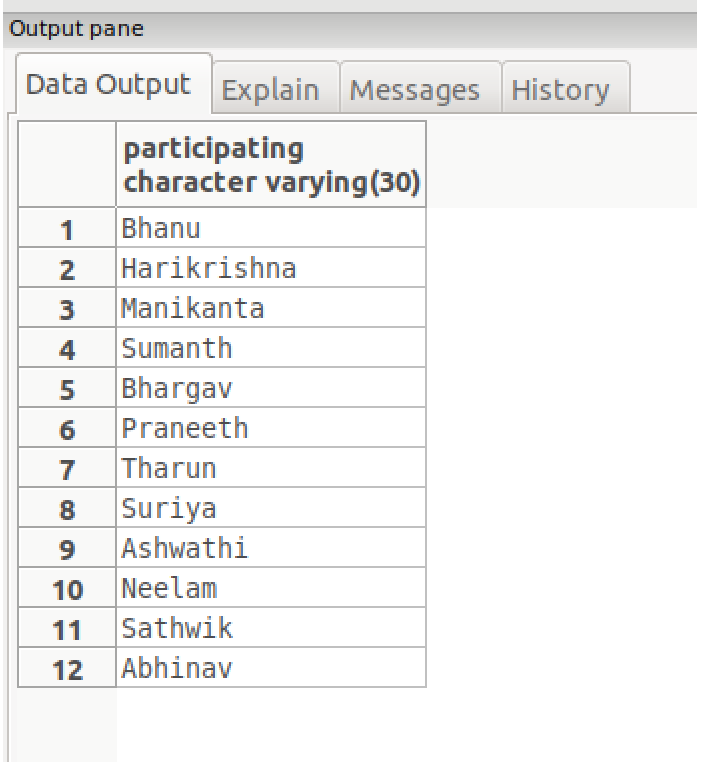
select EventName from Event;



	eventname character varying(20)
1	Dancing Divas
2	Fashion Night
3	Nightingales
4	Face Painting
5	Gaming

List of Participating Students

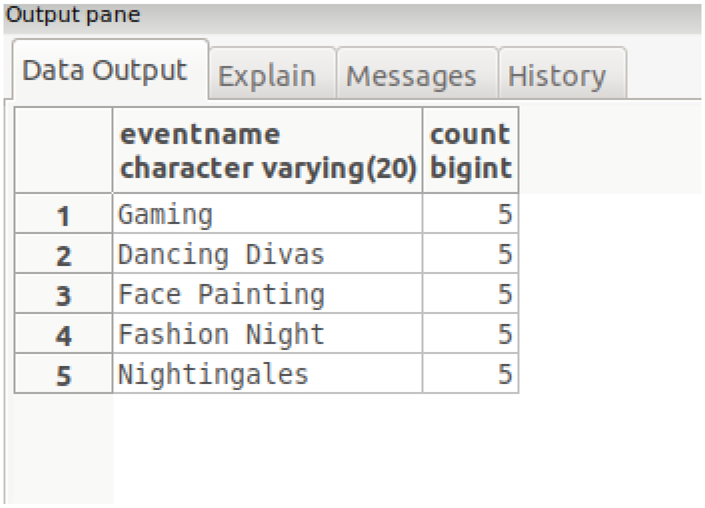
select StudentName as Participating from Student where exists(select * from (Student natural join Participates) as A where Student.StudentID = A.StudentID)



	participating character varying(30)
1	Bhanu
2	Harikrishna
3	Manikanta
4	Sumanth
5	Bhargav
6	Praneeth
7	Tharun
8	Suriya
9	Ashwathi
10	Neelam
11	Sathwik
12	Abhinav

No of Students participating for each Event

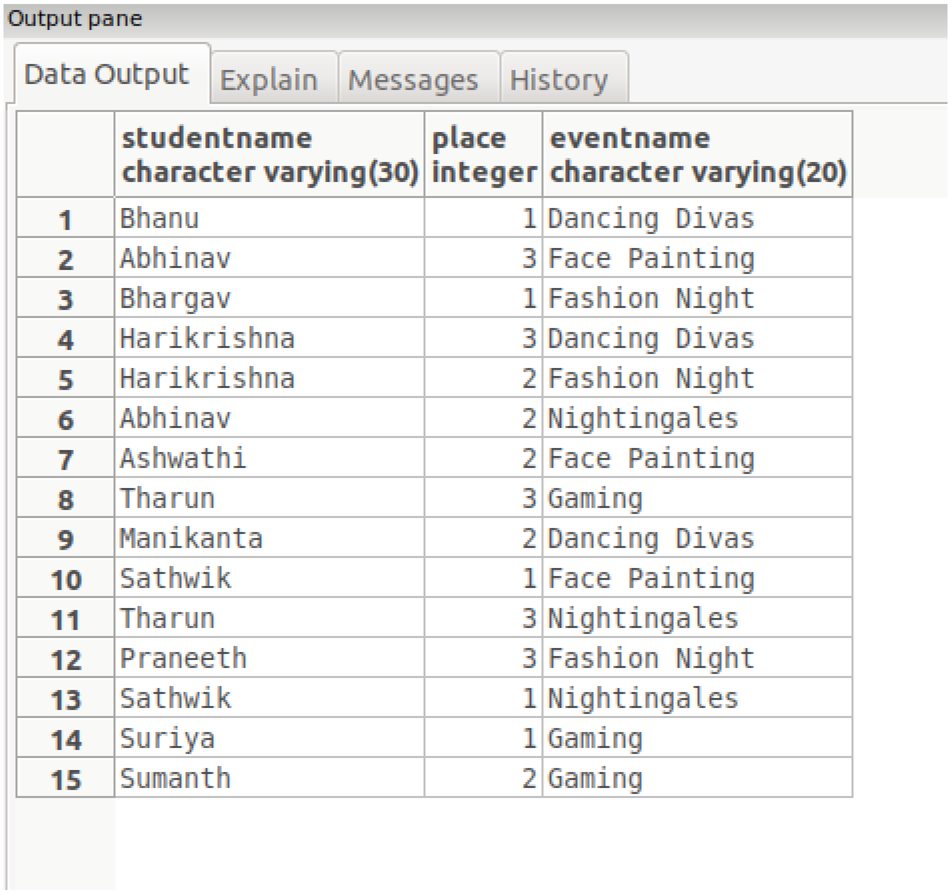
select EventName, count(StudentID) from Participates natural join Event group by EventName;



	eventname character varying(20)	count bigint
1	Gaming	5
2	Dancing Divas	5
3	Face Painting	5
4	Fashion Night	5
5	Nightingales	5

No. of Students who got top 3 places in any Event

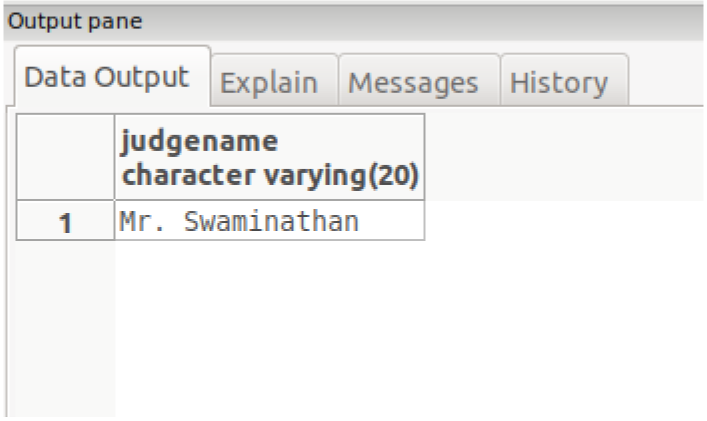
select distinct StudentName, Place, EventName from Student natural join Participates natural join Event where Place > 0;



	studentname	place	eventname
	character varying(30)	integer	character varying(20)
1	Bhanu	1	Dancing Divas
2	Abhinav	3	Face Painting
3	Bhargav	1	Fashion Night
4	Harikrishna	3	Dancing Divas
5	Harikrishna	2	Fashion Night
6	Abhinav	2	Nightingales
7	Ashwathi	2	Face Painting
8	Tharun	3	Gaming
9	Manikanta	2	Dancing Divas
10	Sathwik	1	Face Painting
11	Tharun	3	Nightingales
12	Praneeth	3	Fashion Night
13	Sathwik	1	Nightingales
14	Suriya	1	Gaming
15	Sumanth	2	Gaming

No. of Judges who are Judges in more than 3 events

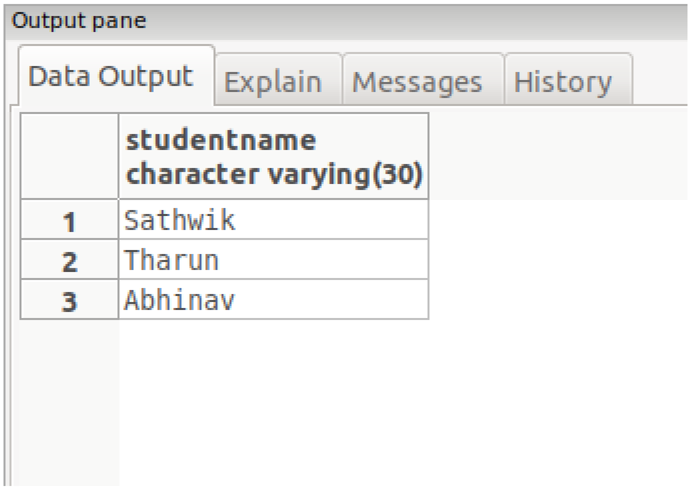
select JudgeName from Judge natural join JudgeEvent group by JudgeID having count(EventID) > 3;



	judgename
	character varying(20)
1	Mr. Swaminathan

List of Students participating in more than 2 events

select StudentName from Student natural join Participates group by StudentID having count(EventID) > 2;



	studentname character varying(30)
1	Sathwik
2	Tharun
3	Abhinav