Database Assignment2

studentID: 1435627

Name: Shu-Han, Yang

Q1: List all events that happened or are scheduled to run in Tasmania and Victoria in the

alphabetical order of event name within each state. Your list should show event code,

name, state and year of the event. Note, you can base event year of the start date of

an event. You need to check how states are stored in the database.

select '1435627' as StuID, event5627.code, event5627.name, eventoccurrence5627.state,

year (eventoccurrence5627.startdate) as year

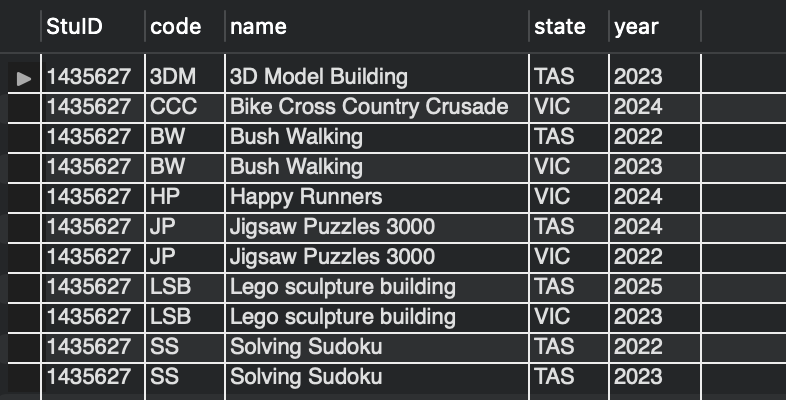
from event5627

inner join eventoccurrence5627

on event5627.code = eventoccurrence5627.eventcode

where eventoccurrence5627.state = 'VIC' or state = 'TAS'

order by event5627.name, eventoccurrence5627.state;



11 rows

Q2: List all events where Danielle Martin was a member of the organizing committee. Your

results should display her staff id, full name in column as last name and first name

separated by a comma (i.e. Martin, Danielle), her role on the committee (e.g. Chair,

Treasurer), name of the event she was organizing, the event occurrence ID and event

occurrence start date. The list should be ordered by event name alphabetically and

event start date from newest to oldest (i.e. from year 2024 down).

select '1435627' as StuID, staff5627.staffID,

concat(staff5627.lastname, ', ' ,staff5627.firstname) as full\_name,

eventcommittee5627.role, event5627.name, eventoccurrence5627.occurrenceID, eventoccurrence5627.startdate

from staff5627

inner join eventcommittee5627

inner join eventoccurrence5627

inner join event5627

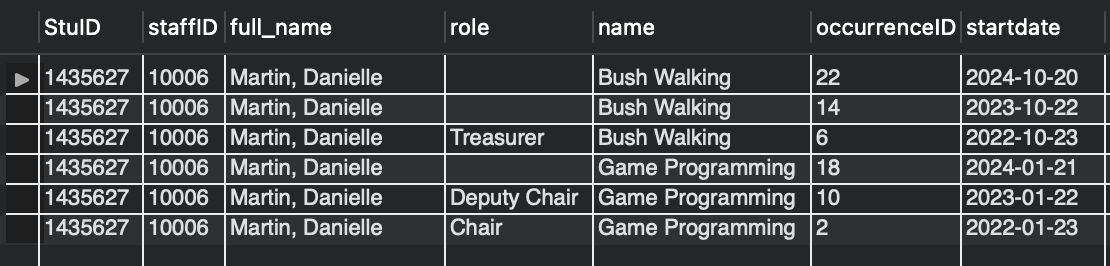
on staff5627.staffID = eventcommittee5627.staffID

and eventcommittee5627.eventoccurrenceID = eventoccurrence5627.occurrenceID

and eventoccurrence5627.eventcode = event5627.code

having full\_name = 'Martin, Danielle'

order by event5627.name, eventoccurrence5627.startdate desc;



6 rows

Q3: How many participants of each gender in each state are stored in the system? Your list

should display State, gender and number of participants in the descending order of

participant numbers within each state.

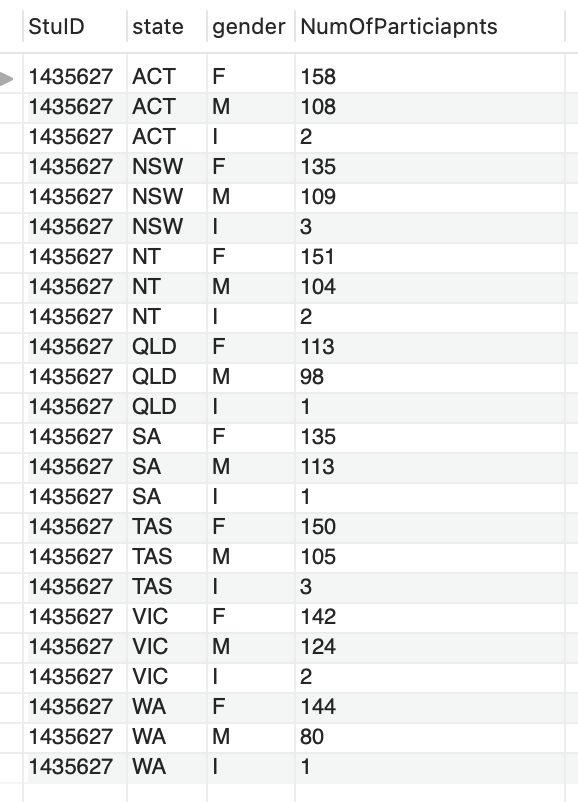
select '1435627' as StuID, participant5627.state, participant5627.gender,

count(participant5627.gender) as NumOfParticiapnts

from participant5627

group by participant5627.state, participant5627.gender

order by participant5627.state, NumOfParticiapnts desc;



24 rows

Q4: Count staff participation in committees, i.e. how many committees each staff member

participated in and list the ones who participated in 6 or more committees. The results

should display staff ID, staff full name as first and last, and number of committees they

participated in, ordered by the number of committees.

select '1435627' as StuID, staff5627.staffID,

concat(staff5627.lastname, ', ' ,staff5627.firstname) as full\_name,

count(eventcommittee5627.eventoccurrenceID) as NumOfCommittees

from staff5627

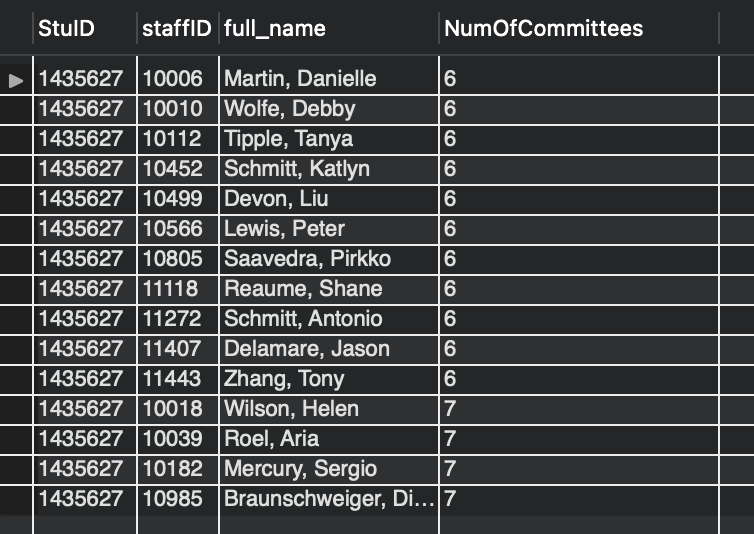
inner join eventcommittee5627

on staff5627.staffID = eventcommittee5627.staffID

group by staff5627.staffID, full\_name

having NumOfcommittees >= 6

order by NumOfcommittees;



15 rows

Q5: List all participants who participated in an event in 2022 but not in any event in 2023. Your list should display participant ID and name and be ordered by participant ID.

select distinct '1435627' as StuID, participant5627.personID, participant5627.name

from participant5627

inner join results5627

inner join eventoccurrence5627

on participant5627.personID = results5627.participant

and results5627.event = eventoccurrence5627.occurrenceID

where year(eventoccurrence5627.startDate) = 2022

having participant5627.personID not in

(select participant5627.personID

from participant5627

inner join results5627

inner join eventoccurrence5627

on participant5627.personID = results5627.participant

and results5627.event = eventoccurrence5627.occurrenceID

where year(eventoccurrence5627.startDate) = 2023 )

order by participant5627.personID;



561 rows

Q6: Count participants in each future event (i.e. participants who signed up for events

scheduled for after the day when we run this query). Your results should show

occurrence number, event name, event start date and number of participants. Events

without signed up participants will show 0. The list should be ordered in the order of

participants numbers from highest to lowest.

select '1435627' as StuID, eventoccurrence5627.occurrenceID, event5627.name,

eventoccurrence5627.startdate,

count(results5627.participant) as NumOfParticiapnts

from eventoccurrence5627

inner join event5627

inner join results5627

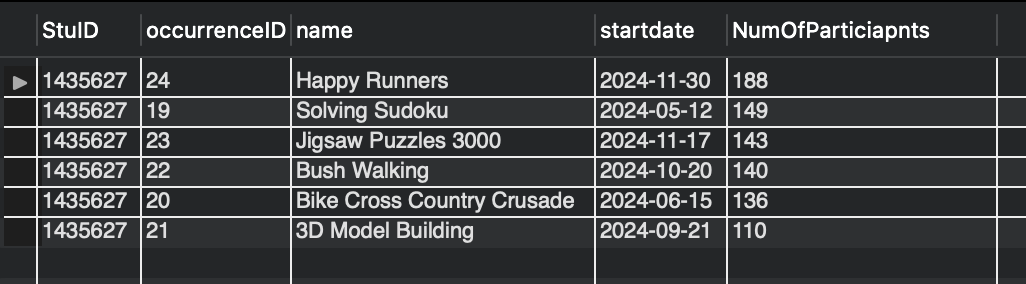
on eventoccurrence5627.eventcode = event5627.code

and eventoccurrence5627.occurrenceID = results5627.event

where eventoccurrence5627.startdate > '2024-5-5'

group by eventoccurrence5627.occurrenceID, event5627.name, eventoccurrence5627.startdate

order by NumOfParticiapnts desc;



6 rows

Q7: List staff members who are chairs of “Solving sudoku” or “Game programming” events.

Your list should show staff full name as a combination of first and last name, email and

role.

select distinct '1435627' as StuID, concat(staff5627.lastname, ', ' ,staff5627.firstname) as full\_name,

staff5627.email, event5627.name, eventcommittee5627.role

from staff5627

inner join eventcommittee5627

inner join eventoccurrence5627

inner join event5627

on staff5627.staffID = eventcommittee5627.staffID

and eventcommittee5627.eventoccurrenceID = eventoccurrence5627.occurrenceID

and eventoccurrence5627.eventcode = event5627.code

where eventcommittee5627.role = 'chair' and event5627.name = 'Solving sudoku'

union

select distinct '1435627' as StuID, concat(staff5627.lastname, ', ' ,staff5627.firstname) as full\_name,

staff5627.email, event5627.name, eventcommittee5627.role

from staff5627

inner join eventcommittee5627

inner join eventoccurrence5627

inner join event5627

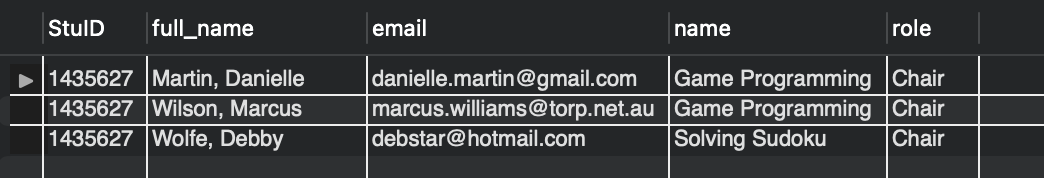
on staff5627.staffID = eventcommittee5627.staffID

and eventcommittee5627.eventoccurrenceID = eventoccurrence5627.occurrenceID

and eventoccurrence5627.eventcode = event5627.code

where eventcommittee5627.role = 'chair' and event5627.name = 'Game programming'

order by full\_name;



3 rows

Q8: Count how many t-shirts of each size was distributed to participants in each event

occurrence. Your results should be ordered by year of the event occurrence, then by t-

shirt size within each event name. The results table should show event occurrence ID,

event year, event name, and number of t-shirts in each size

select distinct '1435627' as StuID, eventoccurrence5627.occurrenceID as event,

year(eventoccurrence5627.startdate) as Event\_year,

event5627.name, participant5627.TShirtSize,

count(participant5627.personID) as No\_Of\_Tshirts

from eventoccurrence5627

inner join event5627

inner join results5627

inner join participant5627

on eventoccurrence5627.eventcode = event5627.code

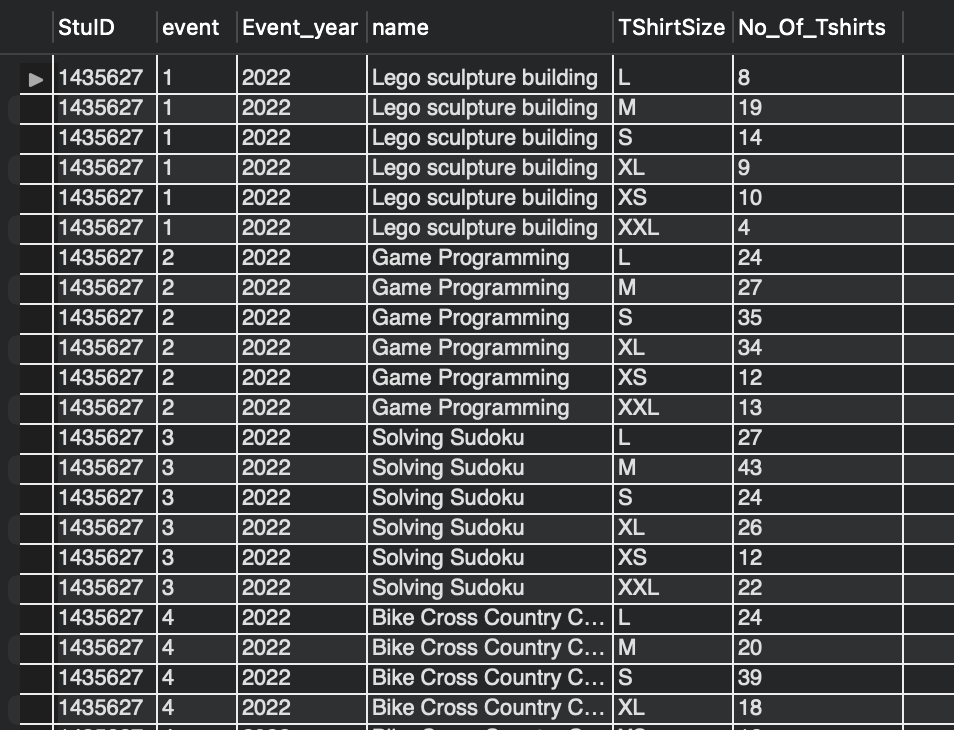
and eventoccurrence5627.occurrenceID = results5627.event

and results5627.participant = participant5627.personID

group by eventoccurrence5627.occurrenceID, year(eventoccurrence5627.startdate) , event5627.name,

participant5627.TShirtSize

order by eventoccurrence5627.occurrenceID, participant5627.TShirtSize;



144 rows

Q9: List participants who took part in all ‘Game Programming’ events that ran in the past 2

years (currently 2022 and 2023, however your query should work in the future). Your

list should show participant ID and name and event name (‘Game Programming’) and

should be ordered by participant name alphabetically.

select '1435627' as StuID, participant5627.personID, participant5627.name, event5627.name as event\_name

from eventoccurrence5627

inner join event5627

inner join results5627

inner join participant5627

on eventoccurrence5627.eventcode = event5627.code

and eventoccurrence5627.occurrenceID = results5627.event

and results5627.participant = participant5627.personID

where event5627.name = 'Game Programming'

and year(eventoccurrence5627.startdate) in (year(now())-1, year(now())-2)

group by participant5627.personID, participant5627.name, event5627.name

having count(participant5627.personID) >= 2

order by participant5627.name;



12 rows

Q10: List all physical events (i.e. involving physical activity as identified by its category) and

associated number of participants over years (e.g. number of participants in all bush

walking events over years). Your list should display event code and name, its category

(involving keyword ‘physical”) and associated number of participants. The results

should be ordered by increasing numbers of participants (i.e. events with lower

number of participants first).

select '1435627' as StuID, event5627.code, event5627.name,

event5627.category, count(participant5627.personID) as NumOfParticiapnts

from event5627

inner join eventoccurrence5627

inner join results5627

inner join participant5627

on eventoccurrence5627.eventcode = event5627.code

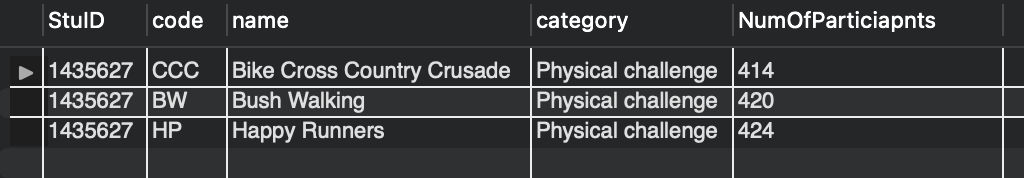
and eventoccurrence5627.occurrenceID = results5627.event

and results5627.participant = participant5627.personID

where event5627.category like 'physical%'

group by event5627.code, event5627.name, event5627.category

order by NumOfParticiapnts;



3 rows

Q11:

a. Write the SQL DDL to create a view that lists age category description and

number of participants in each age category across all events.

Do not include StuID in this view.

In addition to the code, you need to provide 2 screenshots

– the list of tables and views from the left pane of Workbench showing your created view and – the results of running SELECT from your View (we recommend explicitly showing SELECT statement used to create the View).

(10 marks)

b. Using the View you created in Task 10a, list the age group(s) with the lowest

number of participants. Your query needs to display description and the

number of participants. You must include your student ID in task b.

a.

create view my\_view as

select agecategory5627.description, count(participant5627.personID) as NumOfParticiapnts

from agecategory5627

inner join participant5627

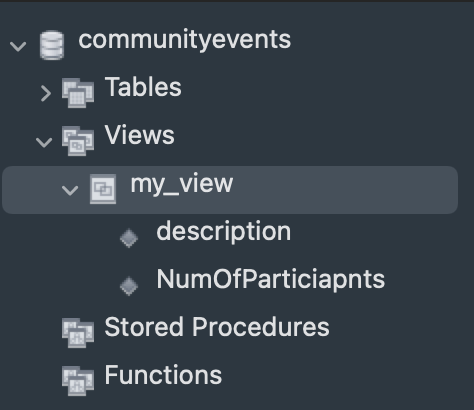
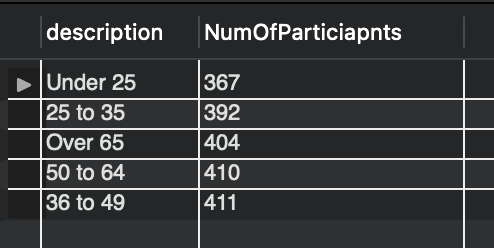
on agecategory5627.catID = participant5627.agecategory

group by agecategory5627.description

order by NumOfParticiapnts;

select \*

from my\_view;

5 rows

b.

select '1435627' as StuID, agecategory5627.description,

count(participant5627.personID) as NumOfParticiapnts

from agecategory5627

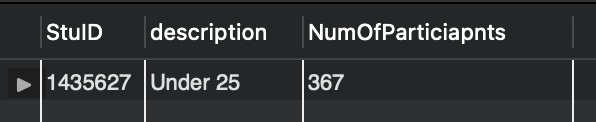
inner join participant5627

on agecategory5627.catID = participant5627.agecategory

group by agecategory5627.description

order by NumOfParticiapnts

limit 1;



1 row