

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;
```

	StuID	code	name	state	year	
▶	1435627	3DM	3D Model Building	TAS	2023	
	1435627	CCC	Bike Cross Country Crusade	VIC	2024	
	1435627	BW	Bush Walking	TAS	2022	
	1435627	BW	Bush Walking	VIC	2023	
	1435627	HP	Happy Runners	VIC	2024	
	1435627	JP	Jigsaw Puzzles 3000	TAS	2024	
	1435627	JP	Jigsaw Puzzles 3000	VIC	2022	
	1435627	LSB	Lego sculpture building	TAS	2025	
	1435627	LSB	Lego sculpture building	VIC	2023	
	1435627	SS	Solving Sudoku	TAS	2022	
	1435627	SS	Solving Sudoku	TAS	2023	

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;
```

	StuID	staffID	full_name	role	name	occurrenceID	startdate
▶	1435627	10006	Martin, Danielle		Bush Walking	22	2024-10-20
	1435627	10006	Martin, Danielle		Bush Walking	14	2023-10-22
	1435627	10006	Martin, Danielle	Treasurer	Bush Walking	6	2022-10-23
	1435627	10006	Martin, Danielle		Game Programming	18	2024-01-21
	1435627	10006	Martin, Danielle	Deputy Chair	Game Programming	10	2023-01-22
	1435627	10006	Martin, Danielle	Chair	Game Programming	2	2022-01-23

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;
```

	StuID	state	gender	NumOfParticiapnts
►	1435627	ACT	F	158
	1435627	ACT	M	108
	1435627	ACT	I	2
	1435627	NSW	F	135
	1435627	NSW	M	109
	1435627	NSW	I	3
	1435627	NT	F	151
	1435627	NT	M	104
	1435627	NT	I	2
	1435627	QLD	F	113
	1435627	QLD	M	98
	1435627	QLD	I	1
	1435627	SA	F	135
	1435627	SA	M	113
	1435627	SA	I	1
	1435627	TAS	F	150
	1435627	TAS	M	105
	1435627	TAS	I	3
	1435627	VIC	F	142
	1435627	VIC	M	124
	1435627	VIC	I	2
	1435627	WA	F	144
	1435627	WA	M	80
	1435627	WA	I	1

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;
```

	StuID	staffID	full_name	NumOfCommittees	
▶	1435627	10006	Martin, Danielle	6	
	1435627	10010	Wolfe, Debby	6	
	1435627	10112	Tipple, Tanya	6	
	1435627	10452	Schmitt, Katlyn	6	
	1435627	10499	Devon, Liu	6	
	1435627	10566	Lewis, Peter	6	
	1435627	10805	Saavedra, Pirkko	6	
	1435627	11118	Reaume, Shane	6	
	1435627	11272	Schmitt, Antonio	6	
	1435627	11407	Delamare, Jason	6	
	1435627	11443	Zhang, Tony	6	
	1435627	10018	Wilson, Helen	7	
	1435627	10039	Roel, Aria	7	
	1435627	10182	Mercury, Sergio	7	
	1435627	10985	Braunschweiger, Di...	7	

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;
```

	StuID	personID	name	
▶	1435627	1006	Beth Meehan	
	1435627	1007	Matthew Parr	
	1435627	1009	Joanne Sharples	
	1435627	1010	Sarah Rankin	
	1435627	1017	Deanna Newman	
	1435627	1023	Blair Habberjam	
	1435627	1024	Madison Romeo	
	1435627	1034	Rebecca Nielsen	
	1435627	1039	Carlos Pineiro	
	1435627	1047	Christopher Hargreaves	
	1435627	1052	Rebecca Croft	
	1435627	1061	Kylie Watson	
	1435627	1065	Renee Reichert	
	1435627	1067	Michael Tobin	
	1435627	1069	Vin Bohnra	
	1435627	1078	Richard Beltiski	
	1435627	1080	Nichola Arnott	

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;
```

	StuID	occurrenceID	name	startdate	NumOfParticiapnts	
►	1435627	24	Happy Runners	2024-11-30	188	
	1435627	19	Solving Sudoku	2024-05-12	149	
	1435627	23	Jigsaw Puzzles 3000	2024-11-17	143	
	1435627	22	Bush Walking	2024-10-20	140	
	1435627	20	Bike Cross Country Crusade	2024-06-15	136	
	1435627	21	3D Model Building	2024-09-21	110	

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;
```

	StuID	full_name	email	name	role	
▶	1435627	Martin, Danielle	danielle.martin@gmail.com	Game Programming	Chair	
	1435627	Wilson, Marcus	marcus.williams@torp.net.au	Game Programming	Chair	
	1435627	Wolfe, Debby	debstar@hotmail.com	Solving Sudoku	Chair	

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;
```

	StuID	event	Event_year	name	TShirtSize	No_Of_Tshirts	
▶	1435627	1	2022	Lego sculpture building	L	8	
	1435627	1	2022	Lego sculpture building	M	19	
	1435627	1	2022	Lego sculpture building	S	14	
	1435627	1	2022	Lego sculpture building	XL	9	
	1435627	1	2022	Lego sculpture building	XS	10	
	1435627	1	2022	Lego sculpture building	XXL	4	
	1435627	2	2022	Game Programming	L	24	
	1435627	2	2022	Game Programming	M	27	
	1435627	2	2022	Game Programming	S	35	
	1435627	2	2022	Game Programming	XL	34	
	1435627	2	2022	Game Programming	XS	12	
	1435627	2	2022	Game Programming	XXL	13	
	1435627	3	2022	Solving Sudoku	L	27	
	1435627	3	2022	Solving Sudoku	M	43	
	1435627	3	2022	Solving Sudoku	S	24	
	1435627	3	2022	Solving Sudoku	XL	26	
	1435627	3	2022	Solving Sudoku	XS	12	
	1435627	3	2022	Solving Sudoku	XXL	22	
	1435627	4	2022	Bike Cross Country C...	L	24	
	1435627	4	2022	Bike Cross Country C...	M	20	
	1435627	4	2022	Bike Cross Country C...	S	39	
	1435627	4	2022	Bike Cross Country C...	XL	18	

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;
```

	StuID	personID	name	event_name	
▶	1435627	2585	Christina Renowden	Game Programming	
	1435627	2527	Claire De Niese	Game Programming	
	1435627	1591	Craig Crouch	Game Programming	
	1435627	2209	Emily Cotugno	Game Programming	
	1435627	2827	Emily Mead	Game Programming	
	1435627	2073	Jamie Fry	Game Programming	
	1435627	1224	Jo Hutchinson	Game Programming	
	1435627	1058	Lauren Gavin	Game Programming	
	1435627	2626	Louise Harfield	Game Programming	
	1435627	1976	Magnus Tan	Game Programming	
	1435627	1805	Max Borowiec	Game Programming	
	1435627	2694	Sarah Irwin	Game Programming	

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;
```

	StuID	code	name	category	NumOfParticiapnts	
▶	1435627	CCC	Bike Cross Country Crusade	Physical challenge	414	
	1435627	BW	Bush Walking	Physical challenge	420	
	1435627	HP	Happy Runners	Physical challenge	424	

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).

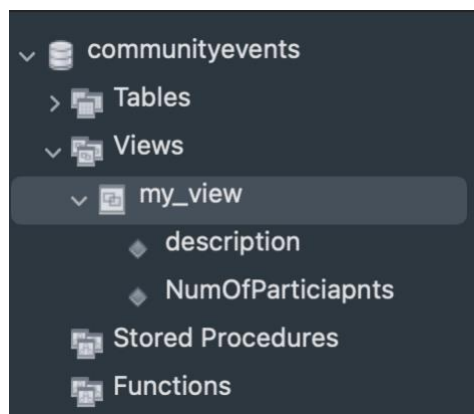
(10 marks)

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;
```



	description	NumOfParticiapnts
▶	Under 25	367
	25 to 35	392
	Over 65	404
	50 to 64	410
	36 to 49	411

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;
```

	StuID	description	NumOfParticiapnts	
▶	1435627	Under 25	367	

1 row