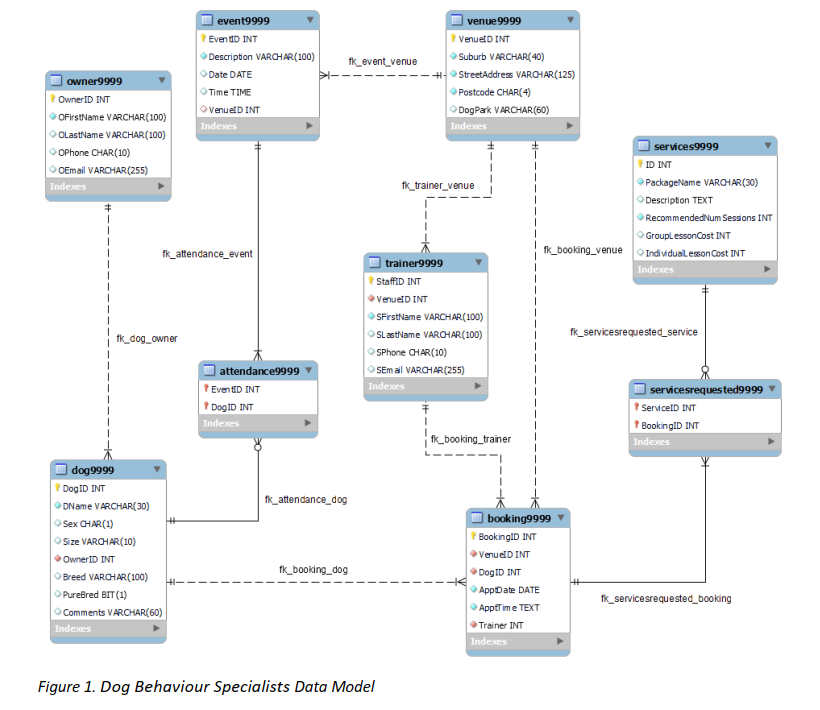
# Dog Behaviour Specialists

Dog Behaviour Specialists is a medium size business operating at several venues positioned near dog parks. Every staff member is associated with a specific venue where they provide dog training services from puppy school basics to addressing behavioural issues to running events for dogs and owners to socialise.



***Note:*** *Run the script on your local MySQL server in MySQL Workbench. This will create*

*the Dog Behaviour Specialists database with all required tables and populate them with data.*

*‘9999’ in all data tables is replaced by last 4 digits of my student id i.e. ‘8875’.*

*Run ‘A2\_2023s2\_DogTraining\_ENG’for database creation.*

**1.** List all events that were conducted at the venues in the suburbs starting with A. Your results should show Suburb and postcode of the venue, as well as description and date of the event. Order the results alphabetically by suburb and by event date within each suburb. If your results show more than one suburb, scroll down to make a screenshot that includes at least 2 suburbs.

SELECT '1398875' AS StuID, v.Suburb, v.Postcode, e.Description, e.date

FROM event8875 e

INNER JOIN venue8875 v ON e.VenueID = v.venueID

WHERE v.Suburb LIKE 'A%'

ORDER BY v.Suburb, e.date



40 rows

**2.** List all services that has never been requested at the venue located in Keilor Downs. Your list should show package name and individual lesson cost ordered alphabetically by package name.

SELECT '1398875' AS StuID, s.PackageName, s.IndividualLessonCost

FROM services8875 s

WHERE s.ID NOT IN (

SELECT sr.ServiceID

FROM servicesrequested8875 sr

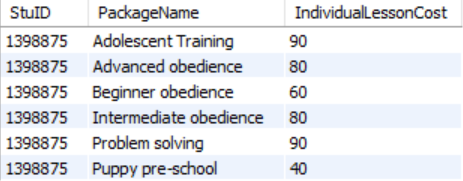
JOIN booking8875 b ON sr.BookingID = b.BookingID

JOIN venue8875 v ON b.VenueID = v.venueID

WHERE v.Suburb = 'Keilor Downs'

)

ORDER BY s.PackageName;



6 rows

**3.** List all events in order of their popularity (from most popular to least popular). Popularity is measured by the number of attendees (dogs) across all venues. So the list should show event description and corresponding total number of attending dogs in the order of highest to lowest numbers.

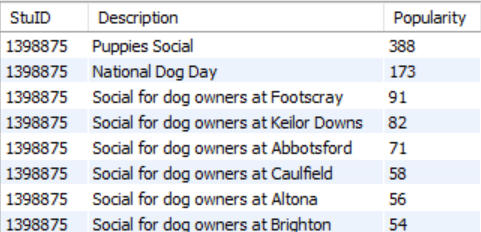
SELECT '1398875' AS StuID, e.Description, COUNT(a.DogID) AS Popularity

FROM event8875 e

INNER JOIN attendance8875 a ON e.EventID = a.EventID

GROUP BY e.Description

ORDER BY Popularity DESC;



8 rows

**4.** List trainers who provided more than 20 sessions/bookings from highest number of sessions to the lowest. Your list should show trainers full name as a combination of first and last name, suburb and number of sessions that trainer provided.

SELECT '1398875' AS StuID, CONCAT(t.SFirstName, ' ', t.SLastName) AS SFullName, v.Suburb, COUNT(b.Trainer) AS NumberOfSessions

FROM trainer8875 t

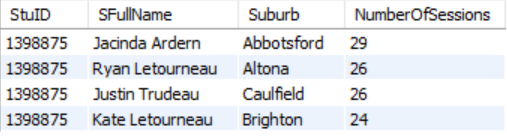
JOIN booking8875 b ON t.StaffID = b.Trainer

JOIN venue8875 v ON t.VenueID = v.venueID

GROUP BY SFullName, v.Suburb

HAVING NumberOfSessions > 20

ORDER BY NumberOfSessions DESC;



4 rows

**5.** List all venues and total for all lessons they provided in the previous year using Group lesson cost data. The report is supposed to include only lessons that offer group mode. Your list should show venue ID, suburb and the total amount in the ascending order of total amounts. Previous year is currently 2022, but the query should be valid whenever it is run in the future.

SELECT '1398875' AS StuID, v.VenueID, v.Suburb, SUM(s.GroupLessonCost) AS TotalAmount

FROM venue8875 v

INNER JOIN booking8875 b ON v.VenueID = b.VenueID

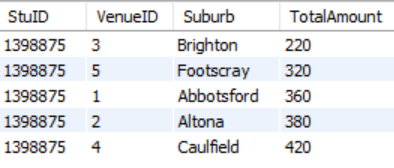
INNER JOIN servicesrequested8875 sr ON b.BookingID = sr.BookingID

INNER JOIN services8875 s ON sr.ServiceID = s.ID

WHERE YEAR(b.ApptDate) = YEAR(CURRENT\_DATE) - 1 AND s.GroupLessonCost IS NOT NULL

GROUP BY v.VenueID, v.Suburb

ORDER BY TotalAmount ASC



5 rows

**6.** List all owners and their dogs who booked any type of obedience lesson (lesson that has the term “obedience” in its package name) at the Caulfield venue. The list should show owner full name as a combination of first and then last name, dog name and package name in the alphabetical order of owner last name.

SELECT DISTINCT '1398875' AS StuID, CONCAT(o.OFirstName, ' ', o.OLastName) AS OFullName, d.DName, s.PackageName

FROM owner8875 o

LEFT OUTER JOIN dog8875 d ON o.ownerID = d.ownerID

LEFT OUTER JOIN booking8875 b ON d.DogID = b.DogID

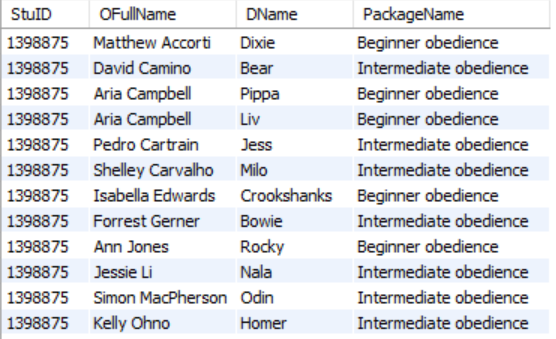
LEFT OUTER JOIN venue8875 v ON b.VenueID = v.VenueID

LEFT OUTER JOIN servicesrequested8875 sr ON b.BookingID = sr.BookingID

LEFT OUTER JOIN services8875 s ON sr.ServiceID = s.ID

WHERE s.PackageName LIKE '%obedience%' AND v.Suburb = 'Caulfield'

ORDER BY o.OLastName ASC



18 rows

**7.** List all owners and their dogs who participated in lessons at Brighton venue in the second quarter of 2023. Second quarter includes April, May, June. The results should display owner full name as first and last, dog name and breed, and lesson date, ordered by lesson date.

SELECT

'1398875' AS StuID,

CONCAT(o.OFirstName, ' ', o.OLastName) AS OFullName,

d.DName,

d.Breed,

b.ApptDate

FROM owner8875 o

INNER JOIN dog8875 d ON o.OwnerID = d.OwnerID

INNER JOIN booking8875 b ON d.DogID = b.DogID

INNER JOIN venue8875 v ON b.VenueID = v.VenueID

WHERE v.Suburb = 'Brighton'

AND b.ApptDate > '2023-03-31'

AND b.ApptDate <= '2023-06-30'

ORDER BY b.ApptDate;

7 rows

**8.** List all owners who attended at least one event or had at least one booking for their dog(s) in January of the previous year, whatever that year might be when the query is run (currently 2022). Your results should show Owner full name (as first and last) and appointment or attendance date. Duplicated records (if any) are allowed.

SELECT '1398875' AS StuID, CONCAT(o.OFirstName, ' ', o.OLastName) AS OFullName, e.Date AS AppAttDate

FROM owner8875 o

INNER JOIN dog8875 d ON o.OwnerID = d.OwnerID

INNER JOIN attendance8875 a ON d.DogID = a.DogID

INNER JOIN event8875 e ON a.EventID = e.EventID

WHERE YEAR(CURRENT\_DATE) - 1 = YEAR(e.Date) AND MONTH(e.Date) = 01

UNION

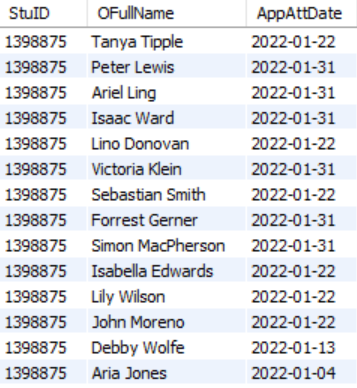
SELECT '1398875' AS StuID, CONCAT(o.OFirstName, ' ', o.OLastName) AS OFullName, b.ApptDate AS AppAttDate

FROM owner8875 o

INNER JOIN dog8875 d ON o.OwnerID = d.OwnerID

INNER JOIN booking8875 b ON d.DogID = b.DogID

WHERE YEAR(CURRENT\_DATE) - 1 = YEAR(b.ApptDate) AND MONTH(b.ApptDate) = 01



17 rows

**9.** Which dogs attended National Dog Day in the past 3 years (i.e. current year, previous year and the one before that)? Your list should display dog name and breed and should be sorted alphabetically by breed, then dog’s name. The query should be useful in the future years so that it shows results to the three years relative to when it is run.

SELECT '1398875' AS StuID, d.DName, d.Breed

FROM dog8875 d

INNER JOIN attendance8875 a ON d.DogID = a.DogID

INNER JOIN event8875 e ON a.EventID = e.EventID

WHERE e.Description LIKE '%National Dog Day%' AND YEAR(e.Date) = YEAR(CURRENT\_DATE) - 2 AND d.DogID IN (

SELECT d1.DogID

FROM dog8875 d1

INNER JOIN attendance8875 a2 ON d1.DogID = a2.DogID

INNER JOIN event8875 e1 ON a2.EventID = e1.EventID

WHERE e1.Description LIKE '%National Dog Day%' AND YEAR(e1.Date) = YEAR(CURRENT\_DATE) - 1 AND d1.DogID IN (

SELECT d2.DogID

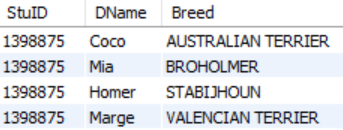
FROM dog8875 d2

INNER JOIN attendance8875 a3 ON d2.DogID = a3.DogID

INNER JOIN event8875 e2 ON a3.EventID = e2.EventID

WHERE e2.Description LIKE '%National Dog Day%' AND YEAR(e2.Date) = YEAR(CURRENT\_DATE)))

ORDER BY d.Breed ASC, d.DName ASC



4 rows

**10. a.** Write the SQL DDL to create a view that lists the dog breed and number of bookings for the service “Advanced Obedience" across all venues. 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).

CREATE VIEW DogBreedBookings AS

SELECT d.Breed, COUNT(\*) AS BookingCount

FROM booking8875 b

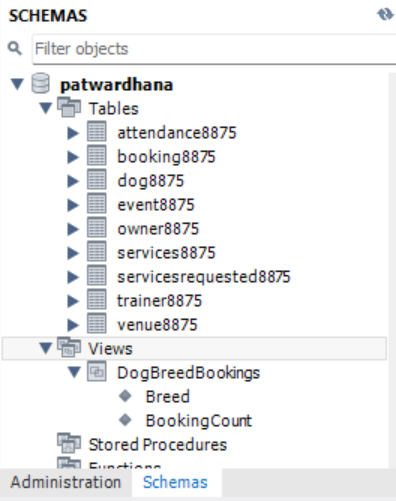
INNER JOIN dog8875 d ON b.DogID = d.DogID

INNER JOIN servicesrequested8875 sr ON b.BookingID = sr.BookingID

INNER JOIN services8875 s ON sr.ServiceID = s.ID

WHERE s.PackageName = 'Advanced Obedience'

GROUP BY d.Breed



SELECT '1398875' AS StuID, d.Breed, COUNT(\*) AS BookingCount

FROM booking8875 b

INNER JOIN dog8875 d ON b.DogID = d.DogID

INNER JOIN servicesrequested8875 sr ON b.BookingID = sr.BookingID

INNER JOIN services8875 s ON sr.ServiceID = s.ID

WHERE s.PackageName = 'Advanced Obedience'

GROUP BY d.Breed



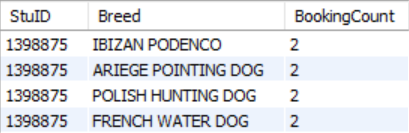
31 rows

**10.b.** Using the View you created in Task 10a, list the breeds with the highest number of bookings. Your query needs to display breed name and the number of lessons booked.

SELECT '1398875' AS StuID, Breed, BookingCount

FROM DogBreedBookings

WHERE BookingCount = (SELECT MAX(BookingCount) FROM DogBreedBookings)



4 rows