

Paw Portal focuses on pet adoption and shelter management, and serves as a system for tracking adoptions, foster care assignments, donations, and medical records for animals. Users can keep track of add/remove pets to the database, find their medical records, view and update information surrounding adopters, track donation data, and perform other operations.

The main change we made to our final schemas was to change all char(20) to varchar(20) in order to keep everything consistent. For example, if Pet has an attribute recordNum of type char(20) and it's a foreign key referencing recordNum in hasMedicalRecord, where it has type varchar(20), then this discrepancy could cause problems.

List of SQL Queries (all found in project.php):

- 1) Insert (function starts line 531)

```
INSERT INTO Pet VALUES (:bind1, :bind2, :bind3, :bind4, :bind5)
```

```
INSERT INTO shelterPet VALUES (:bind1)
```

- 2) Update (function starts line 421)

```
UPDATE Adopter SET " . implode(",", $attributesToUpdate) . " WHERE aid=:aid
```

- 3) Delete (function starts line 590)

```
DELETE FROM hasMedicalRecord WHERE recordNum="" . $recordRow["RECORDNUM"] . ""
```

```
DELETE FROM Pet WHERE pid = :bind1
```

- 4) Selection (function starts line 640)

```
SELECT * FROM Adopter WHERE
```

- 5) Projection (function starts line 670)

```
SELECT $columns FROM fosterParent
```

6) Join (function starts line 827)

```
SELECT P.pName, P.breed, P.species, H.medicalConditions, H.vaccineHistory
FROM Pet P
JOIN hasMedicalRecord H ON P.recordNum = H.recordNum
JOIN shelterPet S ON P.pid = S.pid
WHERE P.$attr = $bindVar
```

7) Aggregation with Group By (function starts from line 899)

```
SELECT D.donationType, AVG(DS.donationAmount) AS avgAmount
FROM Donor D
JOIN Donates DS ON D.did = DS.did
GROUP BY D.donationType
ORDER BY D.donationType
```

This query finds the average donation amount for each type of donation.

8) Aggregation with Having (function starts from line 915)

```
SELECT S.shName, S.shAddress, SUM(Ds.donationAmount) AS totalDonated
FROM Shelter S
JOIN Donates Ds ON S.shAddress = Ds.shAddress
GROUP BY S.shName, S.shAddress
HAVING SUM(Ds.donationAmount) > $min
```

This query finds all the shelters that have received a donation amount greater than a number that the user inputs.

9) Nested Aggregation with Group By (function starts from line 727)

```
SELECT D.shAddress, AVG(D.donationAmount) as avgDonation
FROM Donates D
GROUP BY D.shAddress
HAVING AVG(D.donationAmount) > (SELECT AVG(D2.donationAmount)
                                FROM Donates D2)
```

This query computes the overall average of all donation amounts and finds the shelters that have an average donation amount greater than the overall average.

10) Division (function starts from line 715)

```
SELECT *
FROM Donor D
WHERE NOT EXISTS
    ((SELECT S.shAddress FROM Shelter S)
     MINUS
     (SELECT Ds.shAddress
      FROM Donates Ds
      WHERE Ds.did=D.did))
```

This query finds all the donors that have donated to all the existing shelters.