Definitions - Class 7

Subqueries, IN Conditions, and EXISTS Conditions

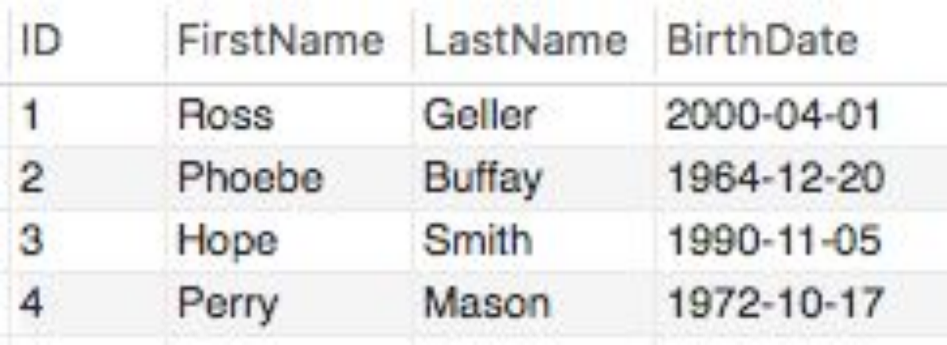
IN Condition

- Allows you to easily test if an expression matches *any* value contained within an entire *list* of values
- Helps to reduce the need for multiple OR conditions.

```
SELECT *

FROM Student

WHERE ID = 1 OR ID = 2 OR ID = 3....
```

IN Condition

```
SELECT *
FROM Student
WHERE ID IN (1, 2, 3, 4);
```

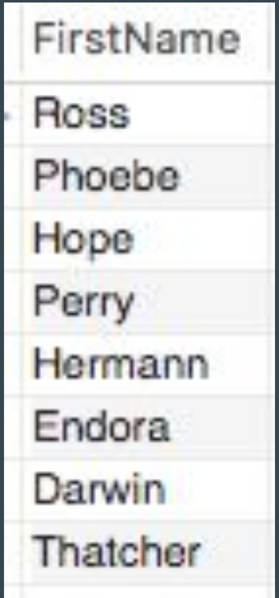


ID	FirstName	LastName	BirthDate
1	Ross	Geller	2000-04-01
2	Phoebe	Buffay	1964-12-20
3	Hope	Smith	1990-11-05
4	Perry	Mason	1972-10-17

Subquery

- A query nested inside a SELECT, INSERT, UPDATE, or DELETE statement (or inside another subquery)
- Can return a set of rows or exactly one row to its parent query

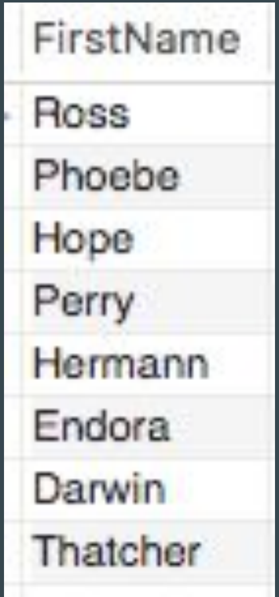
Subquery - Example using IN



FirstName
Ross
Phoebe
Hope
Perry
Hermann
Endora
Darwin
Thatcher

```
SELECT FirstName
FROM Student
WHERE ID IN
    (SELECT StudentID
    FROM ClassStudent
    WHERE ClassID > 1);
```

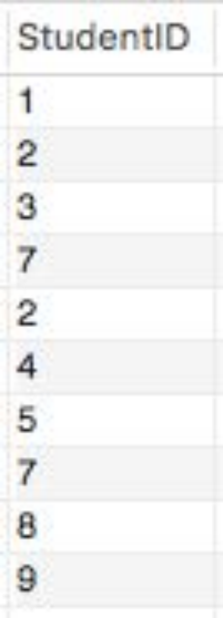
Subquery - Example using IN



FirstName
Ross
Phoebe
Hope
Perry
Hermann
Endora
Darwin
Thatcher

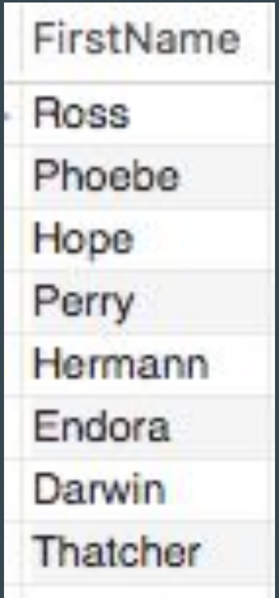
```
SELECT FirstName
FROM Student
WHERE ID IN
    (SELECT StudentID
    FROM ClassStudent
    WHERE ClassID > 1);
```

The subquery returns this result set



StudentID
1
2
3
7
2
4
5
7
8
9

Subquery - Example using IN



FirstName
Ross
Phoebe
Hope
Perry
Hermann
Endora
Darwin
Thatcher

```
SELECT FirstName
FROM Student
WHERE ID IN
        (SELECT StudentID
        FROM ClassStudent
        WHERE ClassID > 1);
```

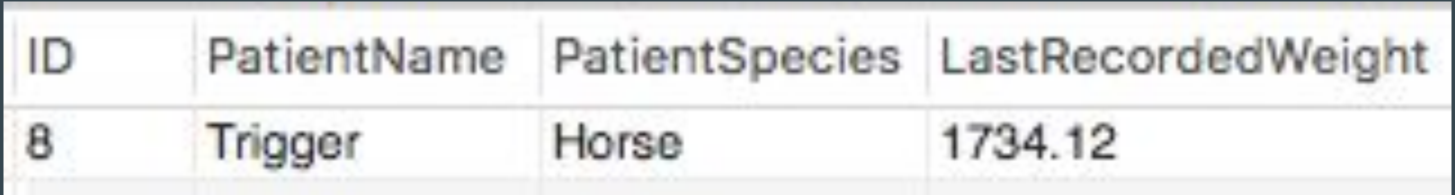
When using IN in a subquery, you can only return a single column in that subquery

Subquery - Scalar Subquery

- A query that returns exactly one value (a single row with a single column)

Subquery - Scalar Subquery

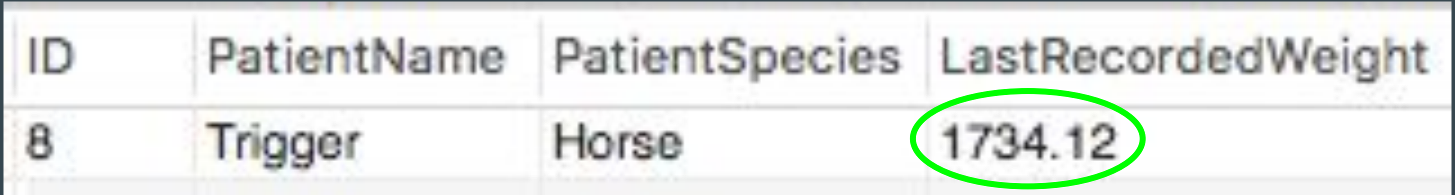
```
SELECT ID, PatientName, PatientSpecies,
LastRecordedWeight

FROM Patient

WHERE LastRecordedWeight =

    (SELECT MAX(LastRecordedWeight) FROM Patient);
```



ID	PatientName	PatientSpecies	LastRecordedWeight
8	Trigger	Horse	1734.12

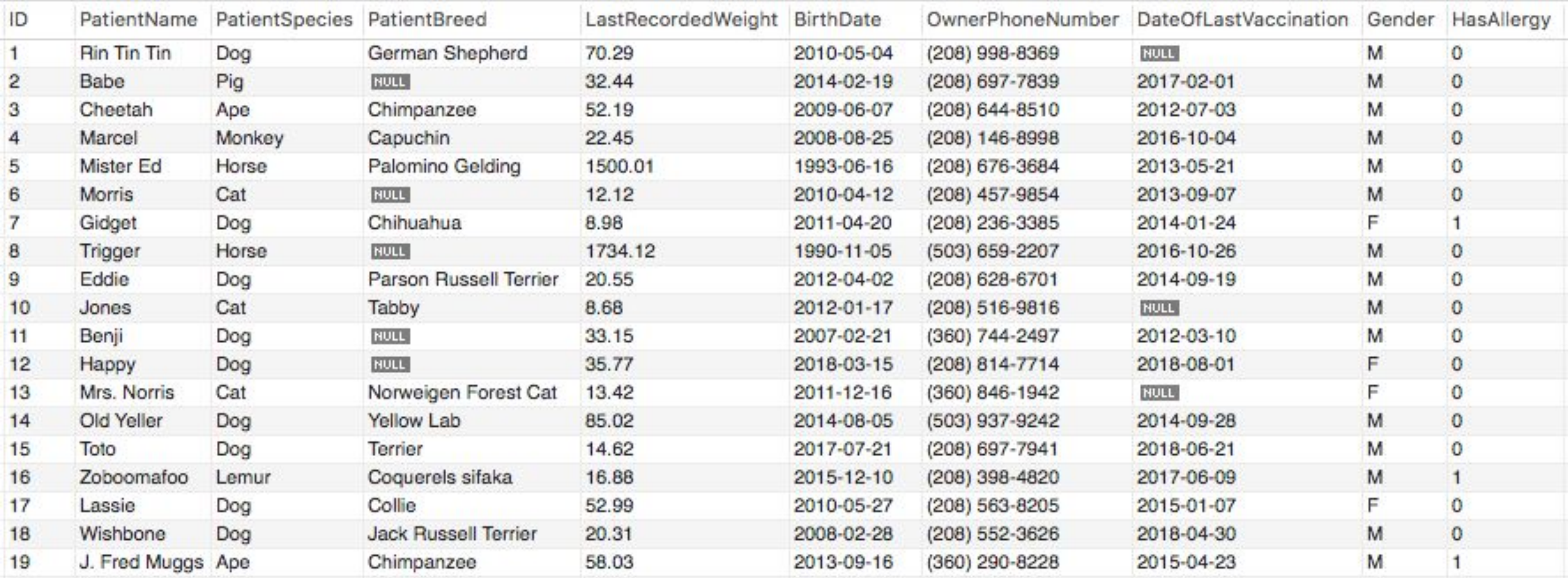
Subquery - Scalar Subquery

```
SELECT ID, PatientName, PatientSpecies,
LastRecordedWeight

FROM Patient

WHERE LastRecordedWeight =

    (SELECT MAX(LastRecordedWeight) FROM Patient);
```



ID	PatientName	PatientSpecies	LastRecordedWeight
8	Trigger	Horse	1734.12

Patient



ID	PatientName	PatientSpecies	PatientBreed	LastRecordedWeight	BirthDate	OwnerPhoneNumber	DateOfLastVaccination	Gender	HasAllergy
1	Rin Tin Tin	Dog	German Shepherd	70.29	2010-05-04	(208) 998-8369	NULL	M	0
2	Babe	Pig	NULL	32.44	2014-02-19	(208) 697-7839	2017-02-01	M	0
3	Cheetah	Ape	Chimpanzee	52.19	2009-06-07	(208) 644-8510	2012-07-03	M	0
4	Marcel	Monkey	Capuchin	22.45	2008-08-25	(208) 146-8998	2016-10-04	M	0
5	Mister Ed	Horse	Palomino Gelding	1500.01	1993-06-16	(208) 676-3684	2013-05-21	M	0
6	Morris	Cat	NULL	12.12	2010-04-12	(208) 457-9854	2013-09-07	M	0
7	Gidget	Dog	Chihuahua	8.98	2011-04-20	(208) 236-3385	2014-01-24	F	1
8	Trigger	Horse	NULL	1734.12	1990-11-05	(503) 659-2207	2016-10-26	M	0
9	Eddie	Dog	Parson Russell Terrier	20.55	2012-04-02	(208) 628-6701	2014-09-19	M	0
10	Jones	Cat	Tabby	8.68	2012-01-17	(208) 516-9816	NULL	M	0
11	Benji	Dog	NULL	33.15	2007-02-21	(360) 744-2497	2012-03-10	M	0
12	Happy	Dog	NULL	35.77	2018-03-15	(208) 814-7714	2018-08-01	F	0
13	Mrs. Norris	Cat	Norweigen Forest Cat	13.42	2011-12-16	(360) 846-1942	NULL	F	0
14	Old Yeller	Dog	Yellow Lab	85.02	2014-08-05	(503) 937-9242	2014-09-28	M	0
15	Toto	Dog	Terrier	14.62	2017-07-21	(208) 697-7941	2018-06-21	M	0
16	Zoboomafoo	Lemur	Coquerels sifaka	16.88	2015-12-10	(208) 398-4820	2017-06-09	M	1
17	Lassie	Dog	Collie	52.99	2010-05-27	(208) 563-8205	2015-01-07	F	0
18	Wishbone	Dog	Jack Russell Terrier	20.31	2008-02-28	(208) 552-3626	2018-04-30	M	0
19	J. Fred Muggs	Ape	Chimpanzee	58.03	2013-09-16	(360) 290-8228	2015-04-23	M	1

Write a Query that Uses a Scalar Subquery

Subquery - Correlated Subquery

- Also referred to as a Synchronized Subquery
- A subquery that references column values from the *outer* query.
- The subquery will be evaluated once for each row processed by the outer query

Example Tables

Patient

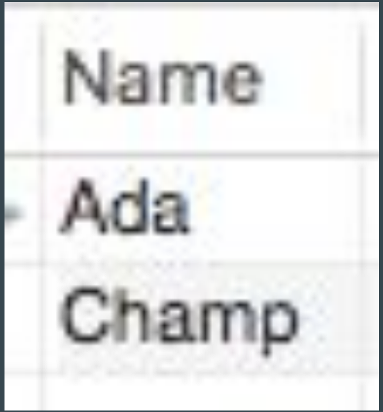
ID	Name	Species	OwnerID
1	Lola	Cat	204
2	Ada	Dog	1
3	Daisy	Dog	33
4	Champ	Ape	98

Vaccination

ID	PatientID	Type	Date
1	3	Rabies	2016-09-02
2	3	Distemper	2016-09-02
3	4	Flu	2014-04-15
4	1	Rabies	2018-08-03

Subquery - Correlated Subquery

```
SELECT p.Name
FROM Patient p
WHERE NOT EXISTS
        (SELECT v.ID
        FROM Vaccination v
        WHERE v.PatientID = p.ID AND v.Type = ‘rabies’ );
```

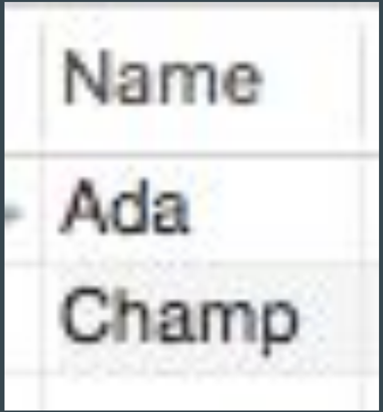


Name
Ada
Champ

Subquery - Correlated Subquery

```
SELECT p.Name
FROM Patient p
WHERE NOT EXISTS
        (SELECT v.ID
        FROM Vaccination v
        WHERE v.PatientID = p.ID AND v.Type = ‘rabies’ );
```

The subquery references the parent table. Hence, the subquery is “correlated”

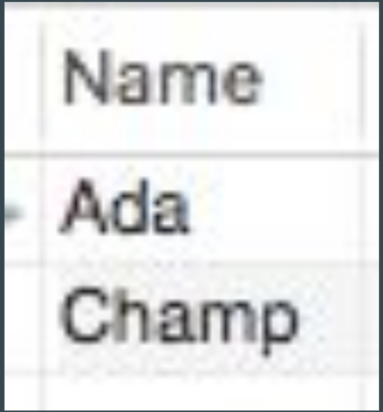


Name
Ada
Champ

Subquery - Correlated Subquery

```
SELECT p.Name
FROM Patient p
WHERE NOT EXISTS
    (SELECT v.ID
    FROM Vaccination v
    WHERE v.PatientID = p.ID AND v.Type = ‘rabies’ );
```

Wait, what’s this??



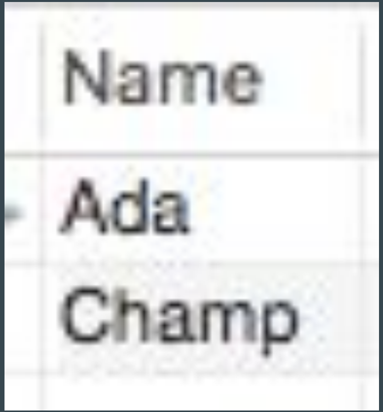
Name
Ada
Champ

EXISTS Condition (A.K.A. EXISTS Operator or EXISTS Statement)

- Often used in correlated subqueries
- Tests for the existence of any set of records in a subquery
- Will assert true if any records in the parent query are in the subquery results set
- You can return multiple columns in the subquery! (This makes it different than IN)

EXISTS Condition (A.K.A. EXISTS Operator or EXISTS Statement)

```
SELECT p.Name
FROM Patient p
WHERE NOT EXISTS
        (SELECT v.ID
        FROM Vaccination v
        WHERE v.PatientID = p.ID AND v.Type = ‘rabies’ );
```

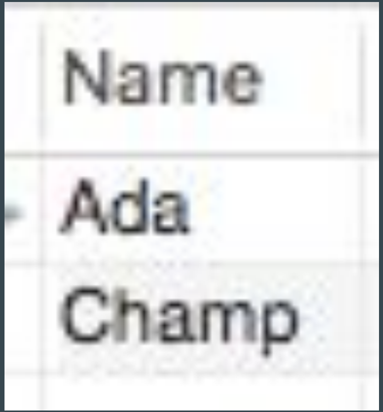


Name
Ada
Champ

EXISTS Condition (A.K.A. EXISTS Operator or EXISTS Statement)

```
SELECT p.Name

FROM Patient p

WHERE NOT EXISTS

      (SELECT v.ID

      FROM Vaccination v

      WHERE v.PatientID = p.ID AND v.Type = ‘rabies’ );
```

This subquery will run for each row in the Patient table



Name
Ada
Champ

EXISTS Condition (A.K.A. EXISTS Operator or EXISTS Statement)

ID 1 (Lola)

```
SELECT v.ID  FROM Vaccination v
WHERE v.PatientID = 1 AND v.Type = ‘rabies’ );
```

EXISTS!

ID 2 (Ada)

```
SELECT v.ID  FROM Vaccination v
WHERE v.PatientID = 2 AND v.Type = ‘rabies’ );
```

DOES NOT EXIST!

ID 3 (Daisy)

```
SELECT v.ID  FROM Vaccination v
WHERE v.PatientID = 3 AND v.Type = ‘rabies’ );
```

EXISTS!

ID 4 (Champ)

```
SELECT v.ID  FROM Vaccination v
WHERE v.PatientID = 4 AND v.Type = ‘rabies’ );
```

DOES NOT

EXIST!

EXISTS Condition (A.K.A. EXISTS Operator or EXISTS Statement)

ID 1 (Lola)

```
SELECT v.ID  FROM Vaccination v
WHERE v.PatientID = 1 AND v.Type = ‘rabies’ );
```

EXISTS!

ID 2 (Ada)

```
SELECT v.ID  FROM Vaccination v
WHERE v.PatientID = 2 AND v.Type = ‘rabies’ );
```

DOES NOT EXIST!

ID 3 (Daisy)

```
SELECT v.ID  FROM Vaccination v
WHERE v.PatientID = 3 AND v.Type = ‘rabies’ );
```

EXISTS!

ID 4 (Champ)

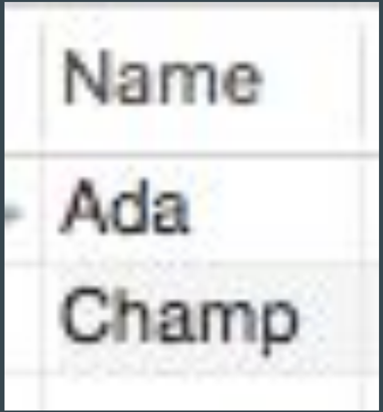
```
SELECT v.ID  FROM Vaccination v
WHERE v.PatientID = 4 AND v.Type = ‘rabies’ );
```

DOES NOT

EXIST!

EXISTS Condition (A.K.A. EXISTS Operator or EXISTS Statement)

```
SELECT p.Name
FROM Patient p
WHERE NOT EXISTS
        (SELECT v.ID
        FROM Vaccination v
        WHERE v.PatientID = p.ID AND v.Type = ‘rabies’ );
```

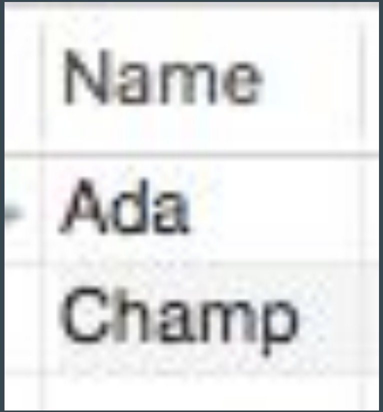


Name
Ada
Champ

EXISTS Condition (A.K.A. EXISTS Operator or EXISTS Statement)

```
SELECT p.Name

FROM Patient p

WHERE NOT EXISTS

    (SELECT v.ID, v.Type

    FROM Vaccination v

    WHERE v.PatientID = p.ID AND v.Type = ‘rabies’ );
```

You can have multiple columns returned in the subquery when using EXISTS or NOT EXISTS



Name
Ada
Champ

Questions?