Aggregations

Wednesday, February 22, 2017

Agenda

- Announcements
- Reading Quiz
- Aggregations Discussion
- 2 Practice Problems
- Group By’s Discussion
- 2 Practice Problems

Announcements

- Heads-up on Lab 3
- Reminder: Complete Lab 3 setup this weekend
- TICKIT demo code: https://github.com/cs327e-spring2017/snippets
- Midterm format

Q1: Which is not an aggregate function?

a) MIN
b) MAX
c) SUM
d) LIKE
e) AVG

Q2: Which statement counts the number of rows in the table Volume?

a) SELECT ROWS (*) from Volume;

b) COUNT (*) from Volume;

c) SELECT COUNT (*) from Volume;

d) ROWS (*) from Volume;

Q3: COUNT(*) includes the records with NULL values.

a) True

b) False

Q4: What is true of aggregate functions?

a) Result of using one of these functions is a computed column that appears only in a result table.

b) They are functions that compute a variety of measures based on values in a column over multiple rows.

c) The basic syntax for these functions is function_name (input_argument).

d) The function call is placed following SELECT.

e) All are true for these functions.

Q5: The GROUP BY clause divides rows into groups that match on one or more values.

a) True

b) False

Standard Aggregate Functions

- MIN
- MAX
- SUM
- AVG
- COUNT

empid	firstname	lastname	depid
1	Michael	Dell	5
2	Betty	Jennings	
3	Bill	Gates	5
4	Fran	Bilas	8

Employee

depid	name
5	Executive
6	Operations
7	Sales
8	Product

Department

Standard Aggregate Functions

- **MIN**
- **MAX**
- **SUM**
- **AVG**
- **COUNT**

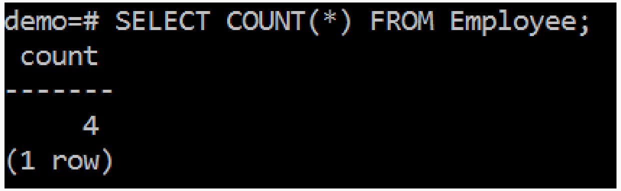
empid	firstname	lastname	depid
1	Michael	Dell	5
2	Betty	Jennings	
3	Bill	Gates	5
4	Fran	Bilas	8

Employee

depid	name
5	Executive
6	Operations
7	Sales
8	Product

Department

```
SELECT COUNT(*) FROM Employee;
```



```
demo=# SELECT COUNT(*) FROM Employee;
 count
-------
     4
(1 row)
```

Standard Aggregate Functions

- MIN
- MAX
- SUM
- AVG
- COUNT

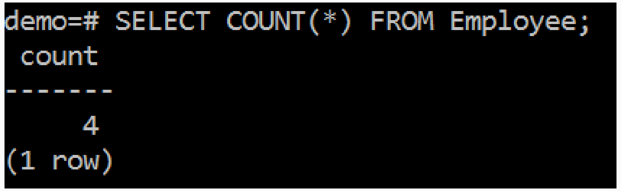
empid	firstname	lastname	depid
1	Michael	Dell	5
2	Betty	Jennings	
3	Bill	Gates	5
4	Fran	Bilas	8

Employee

depid	name
5	Executive
6	Operations
7	Sales
8	Product

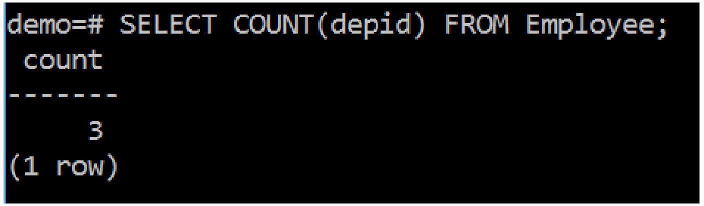
Department

```
SELECT COUNT(*) FROM Employee;
```



```
demo=# SELECT COUNT(*) FROM Employee;
 count
-------
     4
(1 row)
```

```
SELECT COUNT(depid) FROM Employee;
```



```
demo=# SELECT COUNT(depid) FROM Employee;
 count
-------
     3
(1 row)
```

Standard Aggregate Functions

- MIN
- MAX
- SUM
- AVG
- COUNT

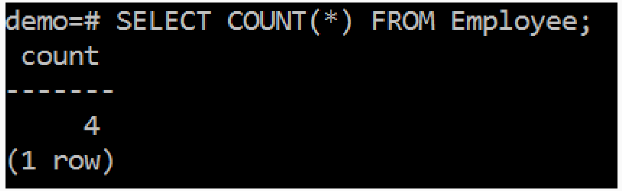
empid	firstname	lastname	depid
1	Michael	Dell	5
2	Betty	Jennings	
3	Bill	Gates	5
4	Fran	Bilas	8

Employee

depid	name
5	Executive
6	Operations
7	Sales
8	Product

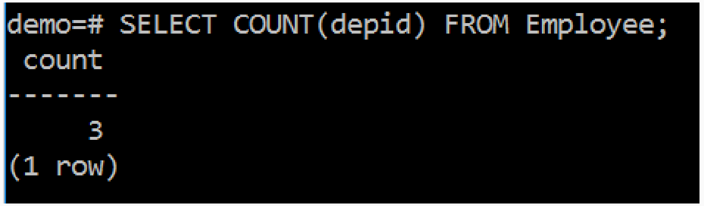
Department

```
SELECT COUNT(*) FROM Employee;
```



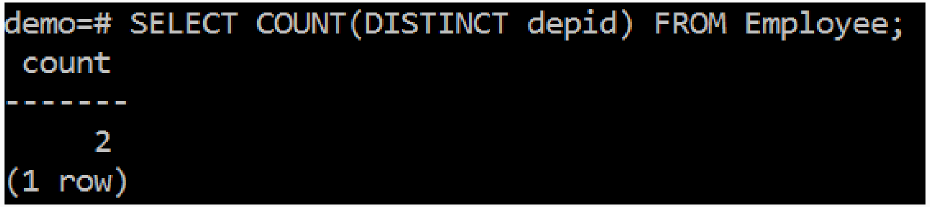
```
demo=# SELECT COUNT(*) FROM Employee;
 count
-------
     4
(1 row)
```

```
SELECT COUNT(depid) FROM Employee;
```



```
demo=# SELECT COUNT(depid) FROM Employee;
 count
-------
     3
(1 row)
```

```
SELECT COUNT(DISTINCT depid) FROM Employee;
```

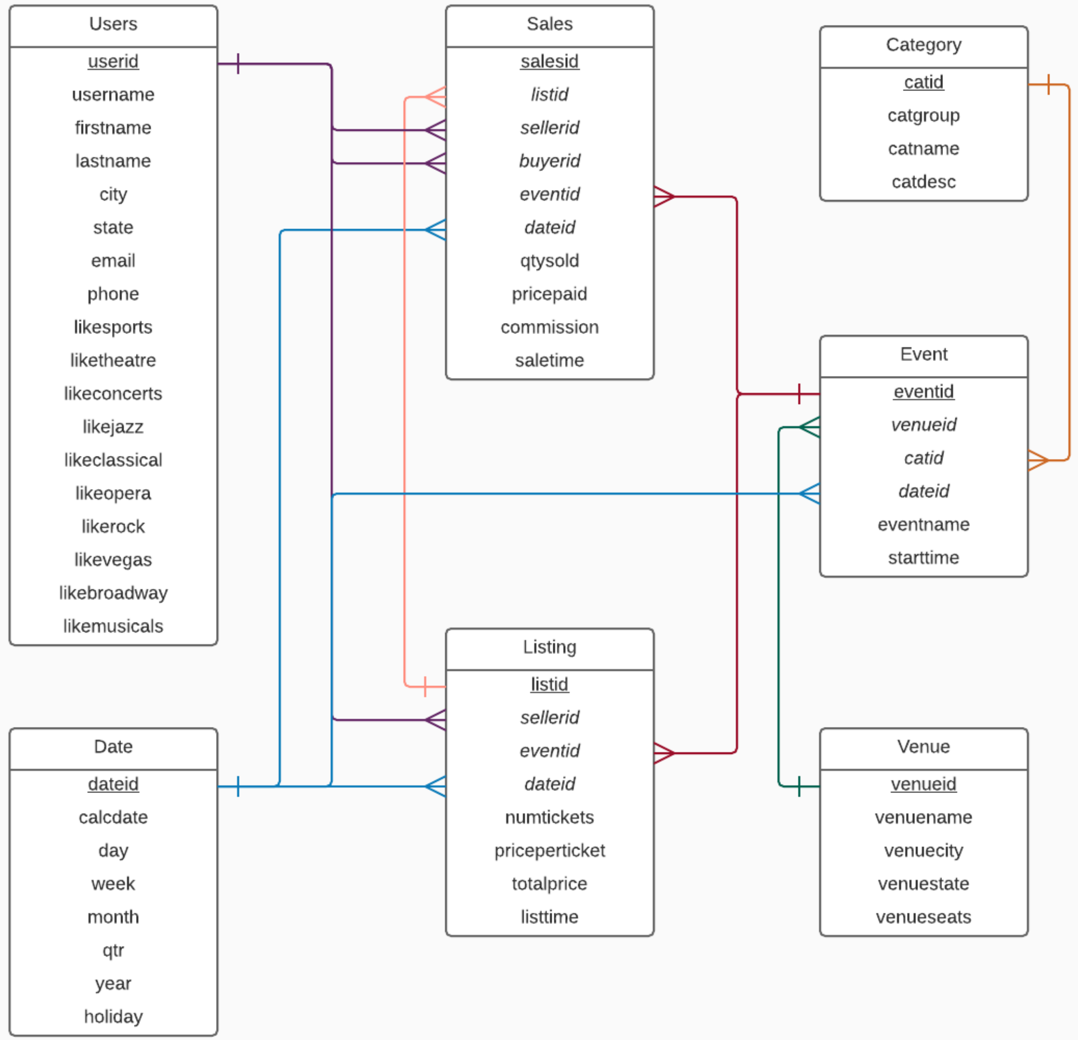


```
demo=# SELECT COUNT(DISTINCT depid) FROM Employee;
 count
-------
     2
(1 row)
```

Practice Problem 1: Calculate the total number of sales, the total quantity of tickets sold and the average sales commission

Notes:

- Use qtysold
- Use commission



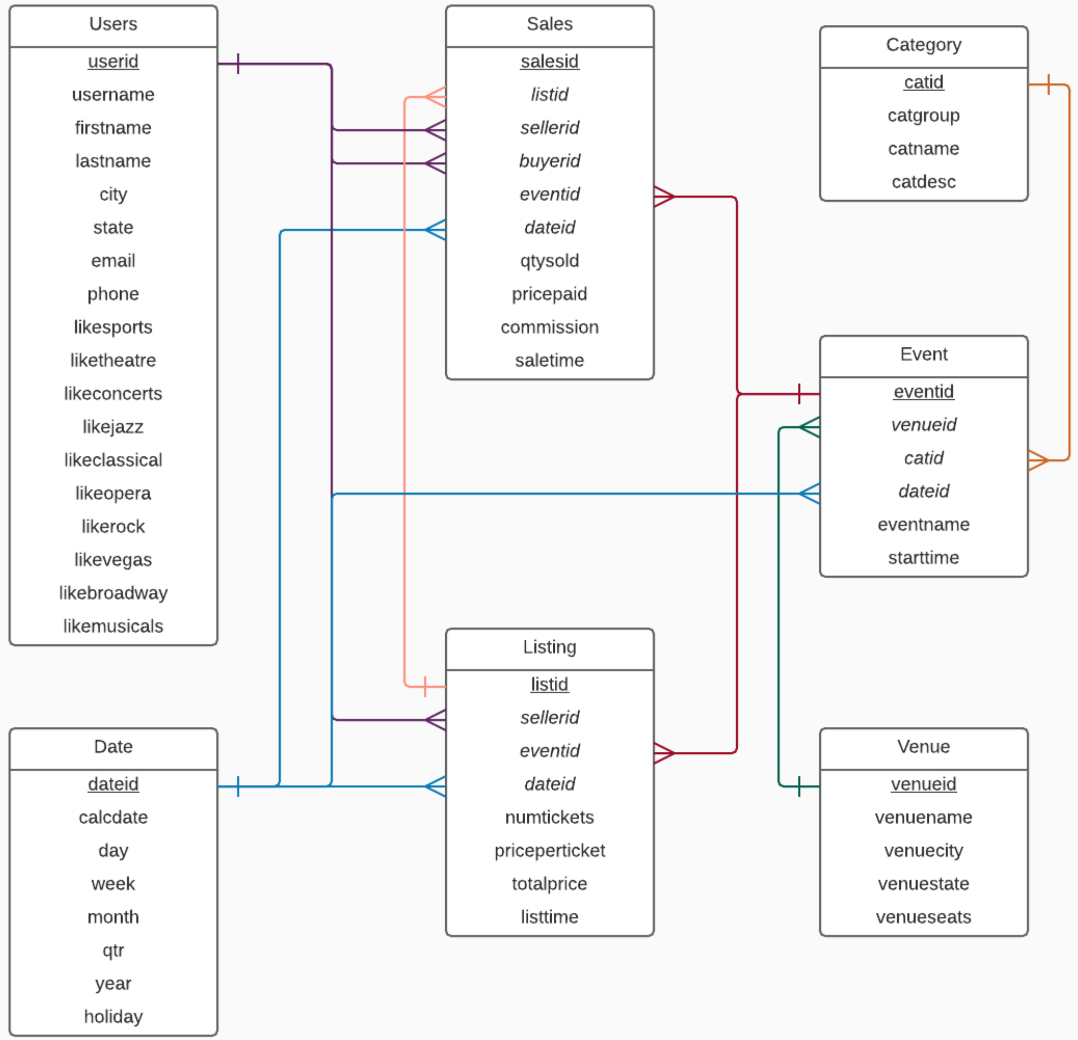
Practice Problem 1: Calculate the total number of sales, the total quantity of tickets sold and the average sales commission

Which aggregate functions are needed to compute the answer?

a) count, sum, avg

b) count, avg

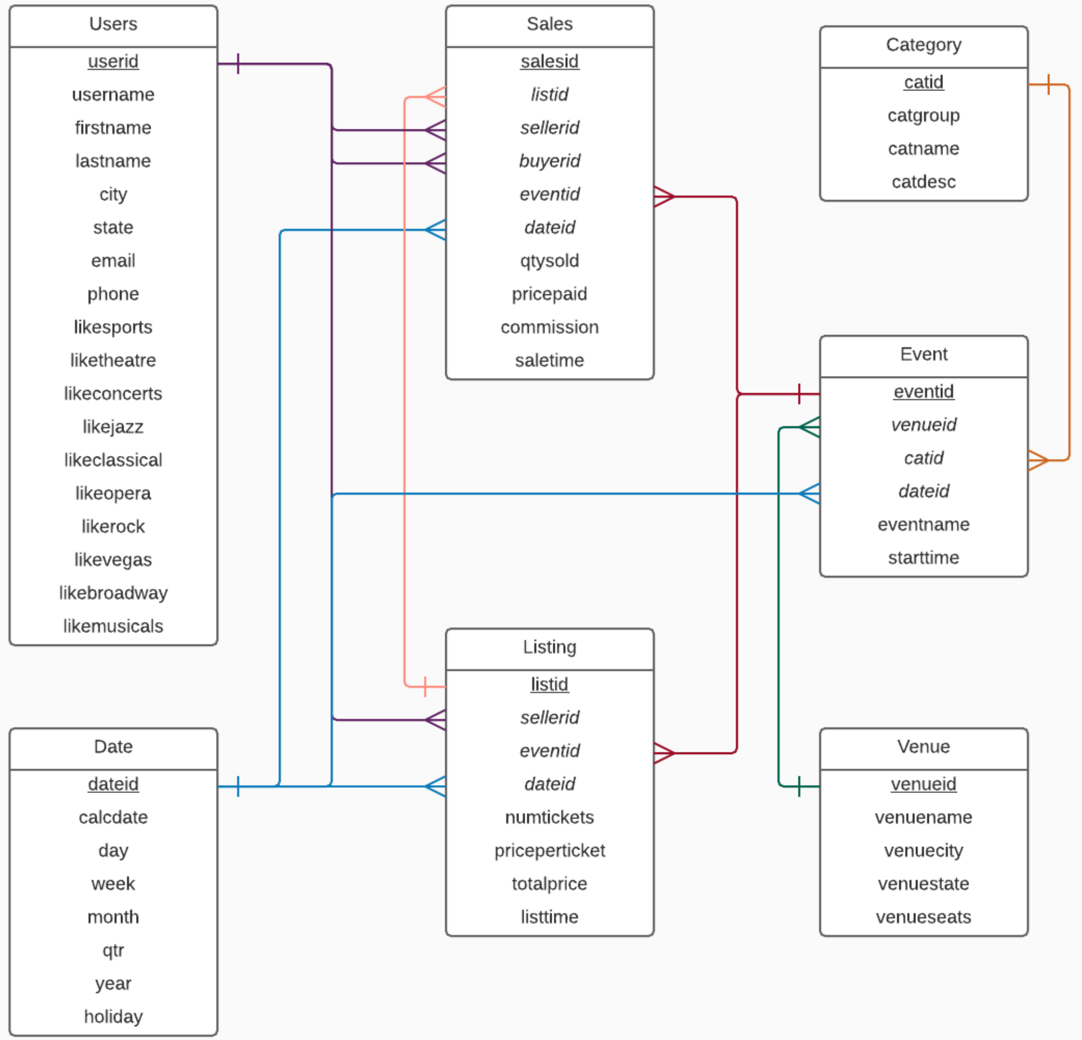
c) sum, avg



Practice Problem 2: Find the lowest and highest price for a 'Spoon' concert ticket

Notes:

- Use priceperticket
- Use eventname = 'Spoon'



Practice Problem 2: Find the lowest and highest price for a 'Spoon' concert ticket

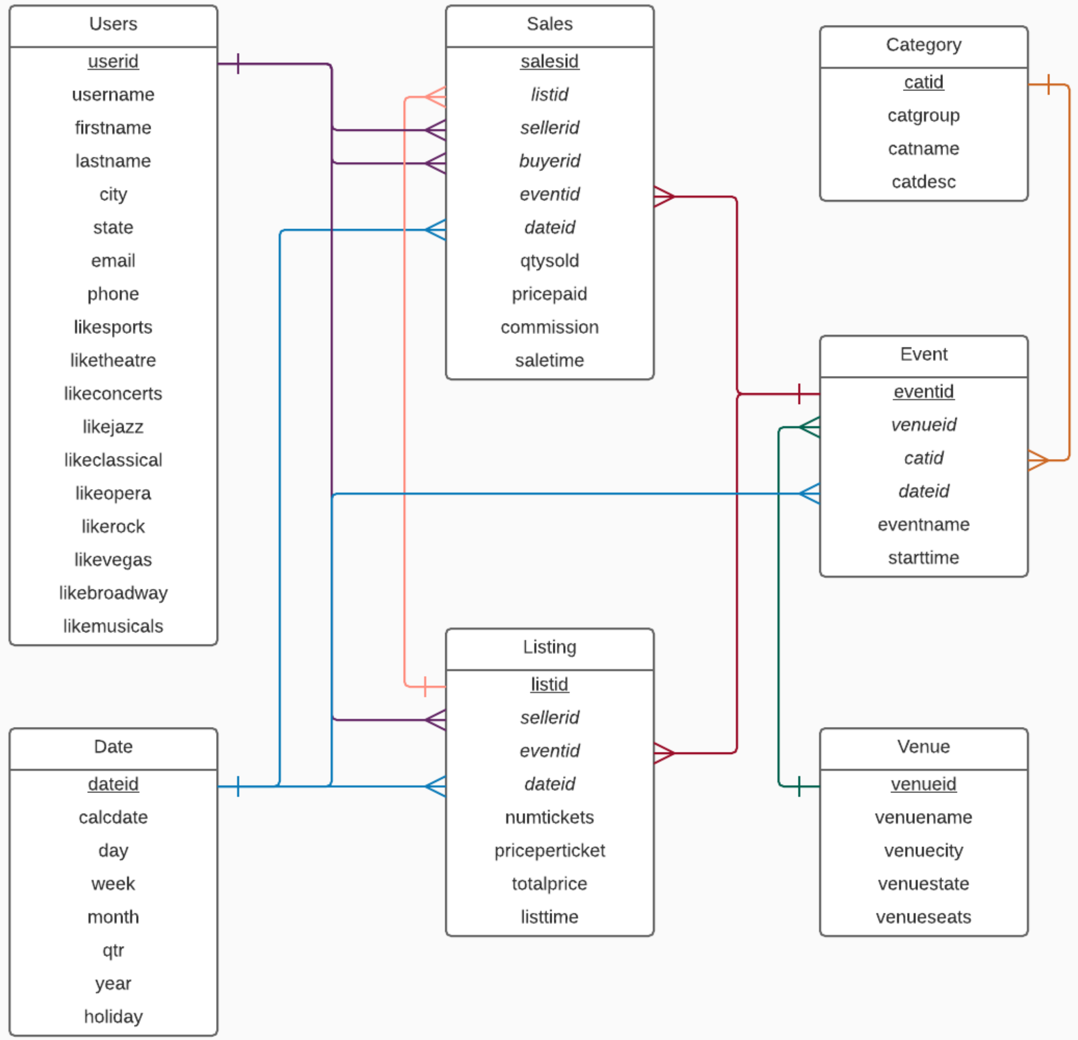
What aggregates are needed to answer this query?

a) min

b) max

c) min, max

d) count, min, max



Aggregates & Groupings

- **MIN**
- **MAX**
- **SUM**
- **AVG**
- **COUNT**

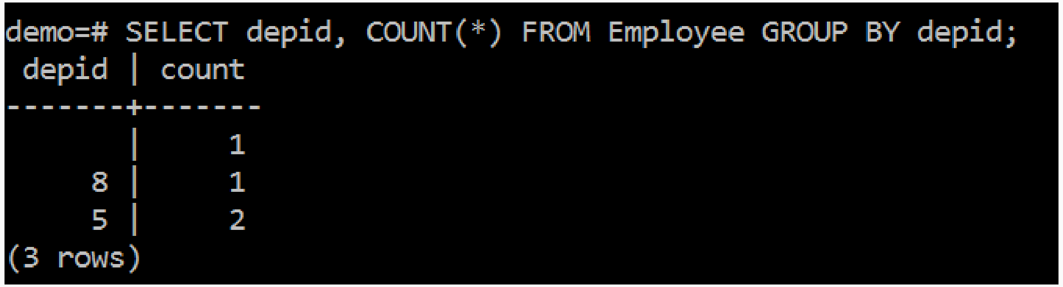
empid	firstname	lastname	depid
1	Michael	Dell	5
2	Betty	Jennings	
3	Bill	Gates	5
4	Fran	Bilas	8

Employee

depid	name
5	Executive
6	Operations
7	Sales
8	Product

Department

```
SELECT depid, COUNT(*) FROM Employee GROUP BY depid;
```



```
demo=# SELECT depid, COUNT(*) FROM Employee GROUP BY depid;
 depid | count
-------+-------
       |     1
     8 |     1
     5 |     2
(3 rows)
```

Aggregates & Groupings

- MIN
- MAX
- SUM
- AVG
- COUNT

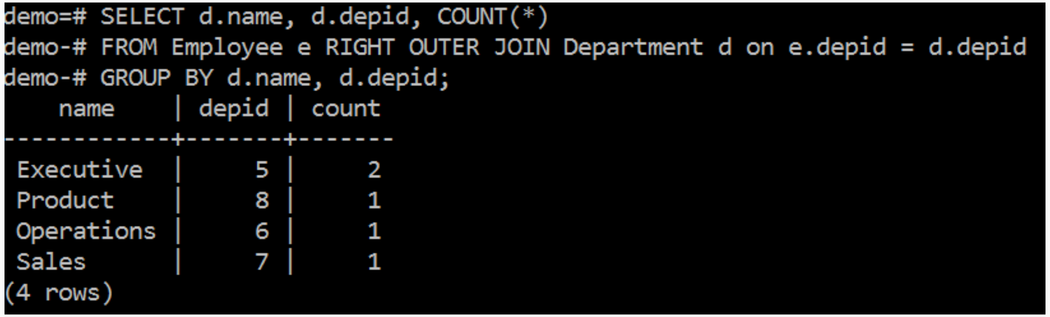
empid	firstname	lastname	depid
1	Michael	Dell	5
2	Betty	Jennings	
3	Bill	Gates	5
4	Fran	Bilas	8

Employee

depid	name
5	Executive
6	Operations
7	Sales
8	Product

Department

```
SELECT d.name, d.depid, COUNT(*)
FROM Employee e RIGHT OUTER JOIN Department d on e.depid = d.depid
GROUP BY d.name, d.depid;
```



```
demo=# SELECT d.name, d.depid, COUNT(*)
demo-# FROM Employee e RIGHT OUTER JOIN Department d on e.depid = d.depid
demo-# GROUP BY d.name, d.depid;
    name    | depid | count
------------+-------+-------
 Executive  |     5 |     2
 Product    |     8 |     1
 Operations |     6 |     1
 Sales      |     7 |     1
(4 rows)
```

Aggregates & Groupings

- MIN
- MAX
- SUM
- AVG
- COUNT

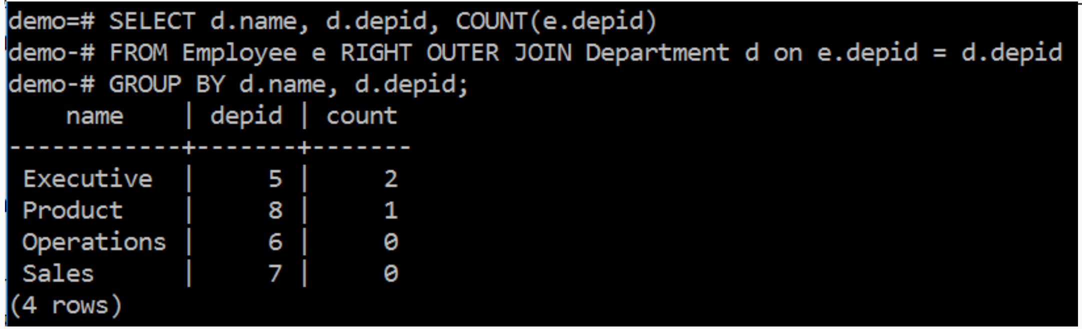
empid	firstname	lastname	depid
1	Michael	Dell	5
2	Betty	Jennings	
3	Bill	Gates	5
4	Fran	Bilas	8

Employee

depid	name
5	Executive
6	Operations
7	Sales
8	Product

Department

```
SELECT d.name, d.depid, COUNT(e.depid)
FROM Employee e RIGHT OUTER JOIN Department d on e.depid = d.depid
GROUP BY d.name, d.depid;
```

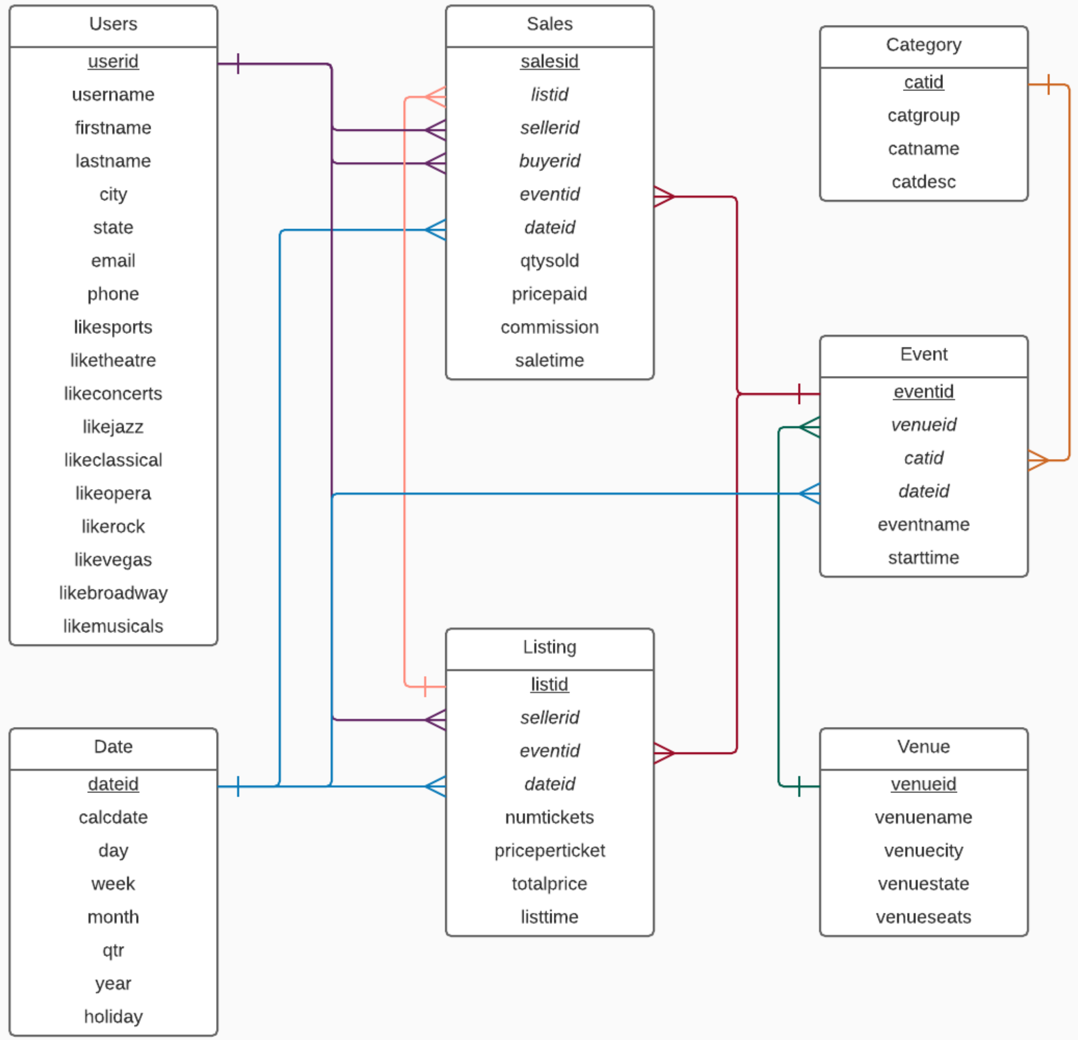


```
demo=# SELECT d.name, d.depid, COUNT(e.depid)
demo-# FROM Employee e RIGHT OUTER JOIN Department d on e.depid = d.depid
demo-# GROUP BY d.name, d.depid;
    name    | depid | count
------------+-------+-------
 Executive  |     5 |     2
 Product    |     8 |     1
 Operations |     6 |     0
 Sales      |     7 |     0
(4 rows)
```

Practice Problem 3: List the categories and the number of events for each one

Notes:

- Use catid and catname for the groupings
- Return catid, catname and the number of events
- Sort the results by catname



Practice Problem 3: List the categories and the number of events for each one

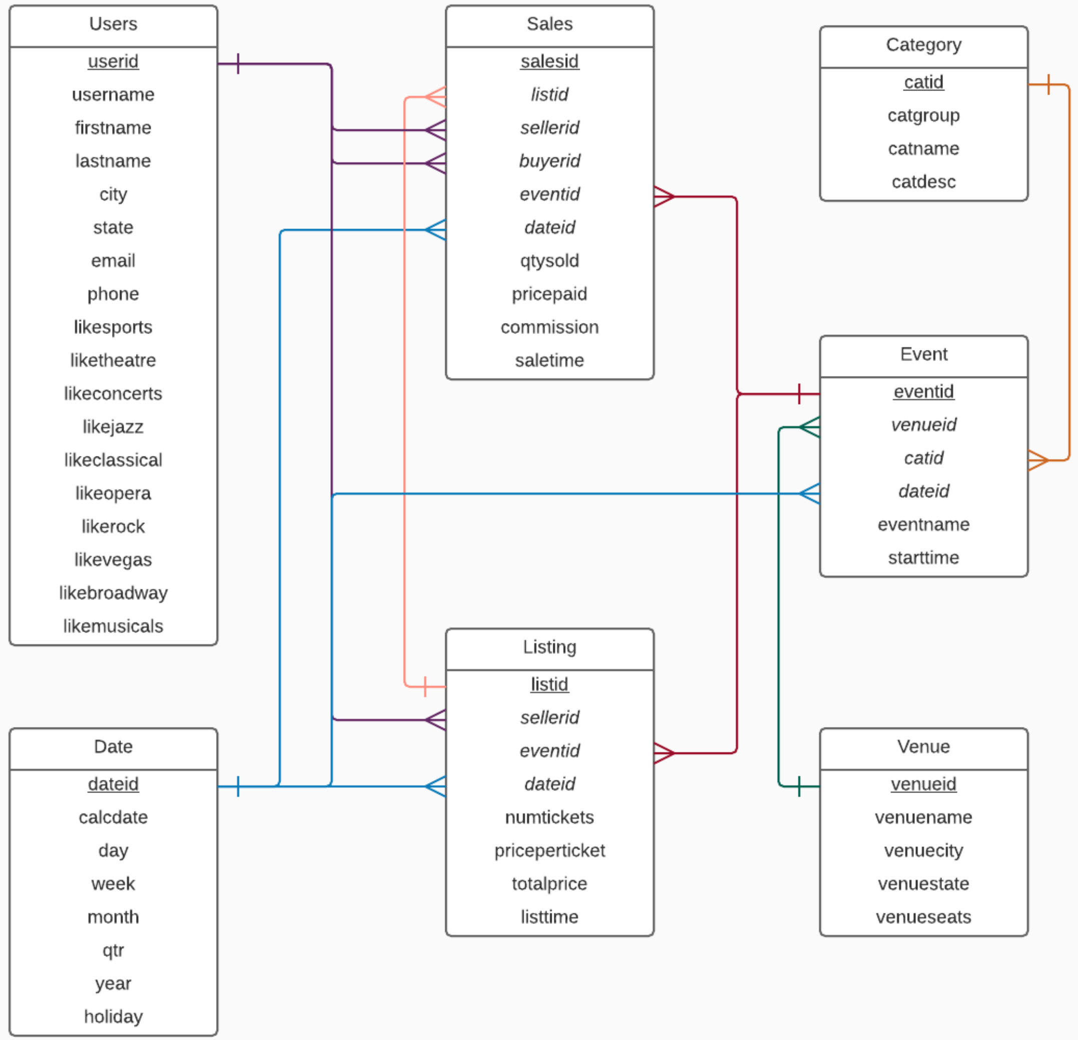
What type of join is needed to answer this query?

a) Inner join

b) Outer join

c) Either one

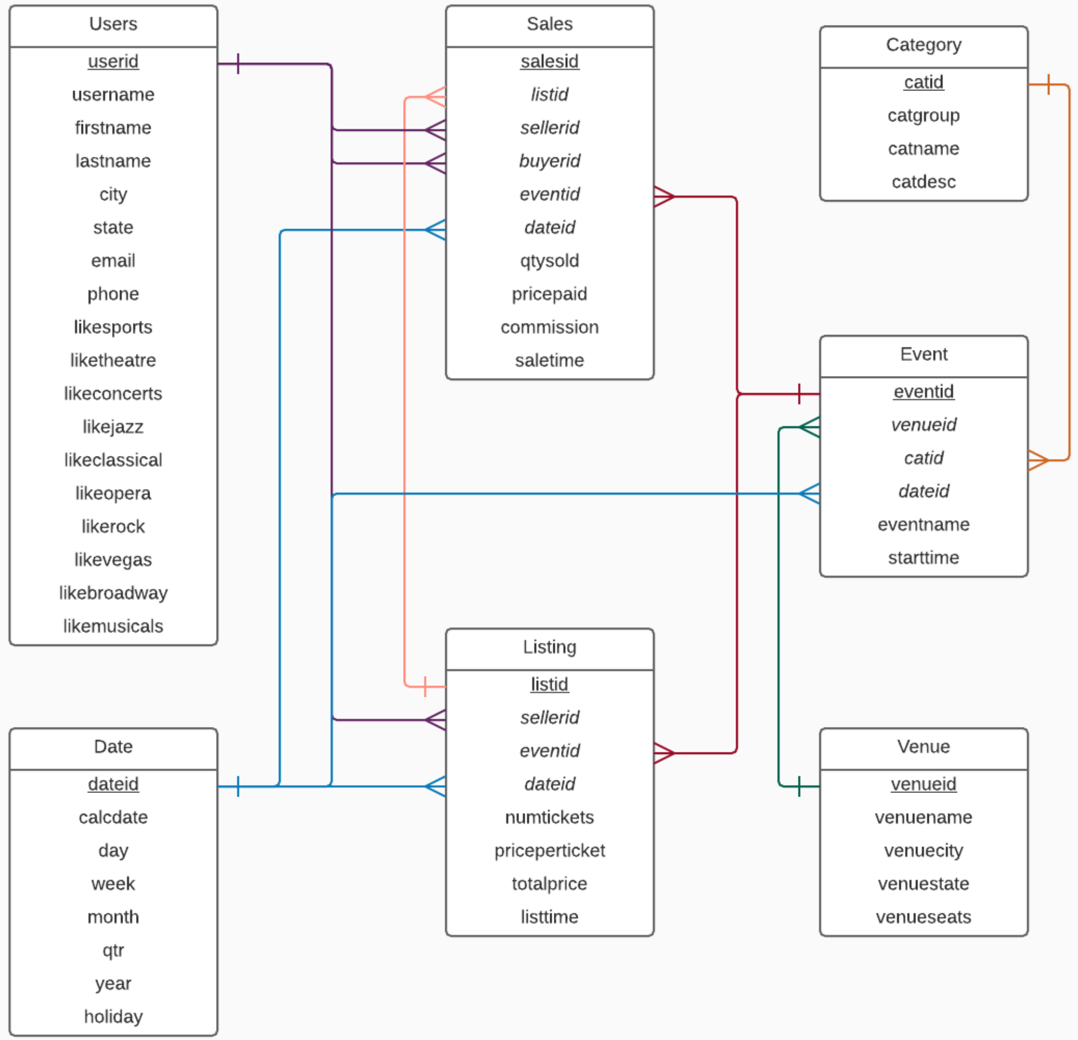
d) Neither one



Practice Problem 4: List the sellers and total commission each earned for 2014 if the commission earned was > 3000

Notes:

- Use sellerid
- Use commission
- Use year = 2014
- Return the sellerid, commission
- Order by commission



Practice Problem 4:

List the sellers and total commission each earned for 2014 if the commission earned was > 3000

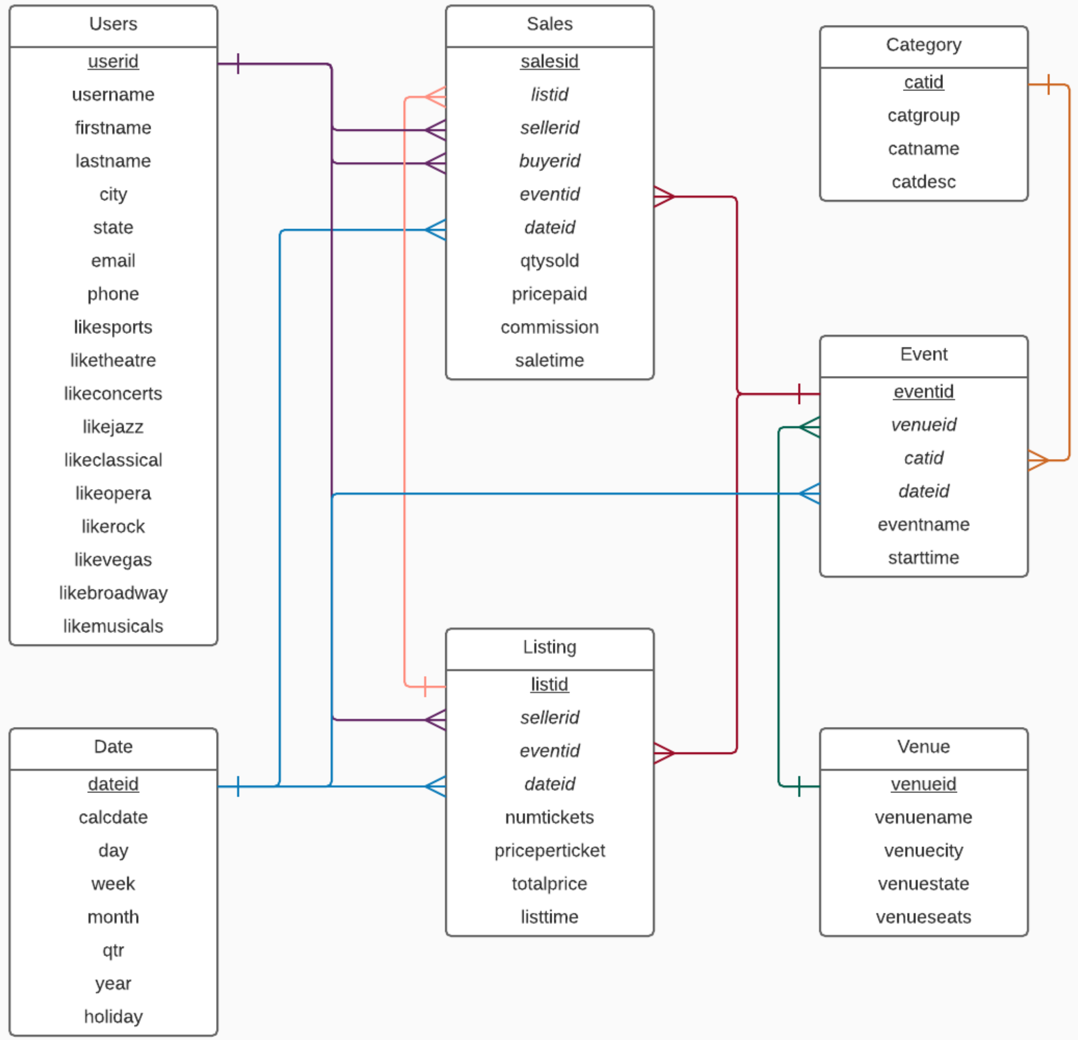
What kind of filter was needed to answer this query?

a) where and having clause

b) where or having clause

c) only where clause

d) only having clause



Practice Problem Solutions

Find solutions to practice problem in our snippets repo:

https://github.com/cs327e-spring2017/snippets (filenames start with “tickit_”)