Aggregate Queries and Views

CS 327E
October 11, 2017

Announcements:

- Midterm: 2 weeks from today in **ETC 2.108**
- Next week: Lab 3

1) The SQL command for adding a new record to an existing table is:

A) ADD
B) PUT
C) INSERT
D) CREATE

2) Given the input tables shown, what will the following statement do?

```
INSERT INTO Employee2
SELECT * FROM Employee;
```

A) Copy 6 records from Employee to Employee2.

B) Copy 6 records from Employee2 to Employee.

C) Copy 0 records.

```
dev=> select * from Employee;
 empid | firstname | lastname | salary | depid
-------+-----------+----------+--------+-------
     1 | Michael   | Dell     |    100 |     5
     2 | Betty     | Jennings |    200 |
     3 | Bill      | Gates    |      0 |     5
     4 | Kay       | McNulty  |    300 |     8
     5 | Jim       | Gray     |    500 |     6
     6 | Gordon    | Moore    |    400 |     6
(6 rows)


dev=> select * from Employee2;
 empid | firstname | lastname | salary | depid
-------+-----------+----------+--------+-------
(0 rows)
```

3) The SQL command for updating a record in a table is:

A) UPDATE
B) ALTER
C) MODIFY
D) CHANGE

4) The SQL command for deleting a record from a table is:

A) TRUNCATE
B) PURGE
C) DROP
D) DELETE

5) Given the input tables shown and the following statement, how many columns will the resulting view have?

```
CREATE VIEW Emp_Dept AS
  SELECT e.*, d.depname
  FROM Employee e
  JOIN Department d
  ON e.depid = d.depid;
```

A) 5
B) 6
C) 7

```
dev=> select * from Employee;
 empid | firstname | lastname | salary | depid
-------+-----------+----------+--------+-------
     1 | Michael   | Dell     |    100 |     5
     2 | Betty     | Jennings |    200 |
     3 | Bill      | Gates    |      0 |     5
     4 | Kay       | McNulty  |    300 |     8
     5 | Jim       | Gray     |    500 |     6
     6 | Gordon    | Moore    |    400 |     6
(6 rows)


dev=> select * from Department;
 depid |   depname
-------+-------------
     5 | Executive
     7 | Sales
     8 | Engineering
     6 | Research
(4 rows)
```

Semantics of COUNT

```
dev=> select count(*) from Employee;
 count
-------
     6
(1 row)
```

```
dev=> select count(depid) from Employee;
 count
-------
     5
(1 row)
```

```
dev=> select count(distinct depid) from Employee;
 count
-------
     3
(1 row)
```

Employee

empid	firstname	lastname	salary	depid
1	Michael	Dell	100	5
2	Betty	Jennings	200	
3	Bill	Gates	0	5
4	Kay	McNulty	300	8
5	Jim	Gray	500	6
6	Gordon	Moore	400	6

What's wrong with this query?

```
dev=> select depname, d.depid, count(*)
dev-> from Employee e full outer join Department d
dev-> on e.depid = d.depid
dev-> group by depname, d.depid;
   depname    | depid | count
--------------+-------+-------
              |       |     1
 Sales        |     7 |     1
 Engineering  |     8 |     1
 Executive    |     5 |     2
 Research     |     6 |     2
(5 rows)
```

Employee

empid	firstname	lastname	salary	depid
1	Michael	Dell	100	5
2	Betty	Jennings	200	
3	Bill	Gates	0	5
4	Kay	McNulty	300	8
5	Jim	Gray	500	6
6	Gordon	Moore	400	6

Department

depid	depname
5	Executive
6	Research
7	Sales
8	Engineering

This is what we want:

```
dev=> select depname, d.depid, count(empid)
dev-> from Employee e full outer join Department d
dev-> on e.depid = d.depid
dev-> group by depname, d.depid;
   depname   | depid | count
-------------+-------+-------
             |       |     1
 Sales       |     7 |     0
 Engineering |     8 |     1
 Executive   |     5 |     2
 Research    |     6 |     2
(5 rows)
```

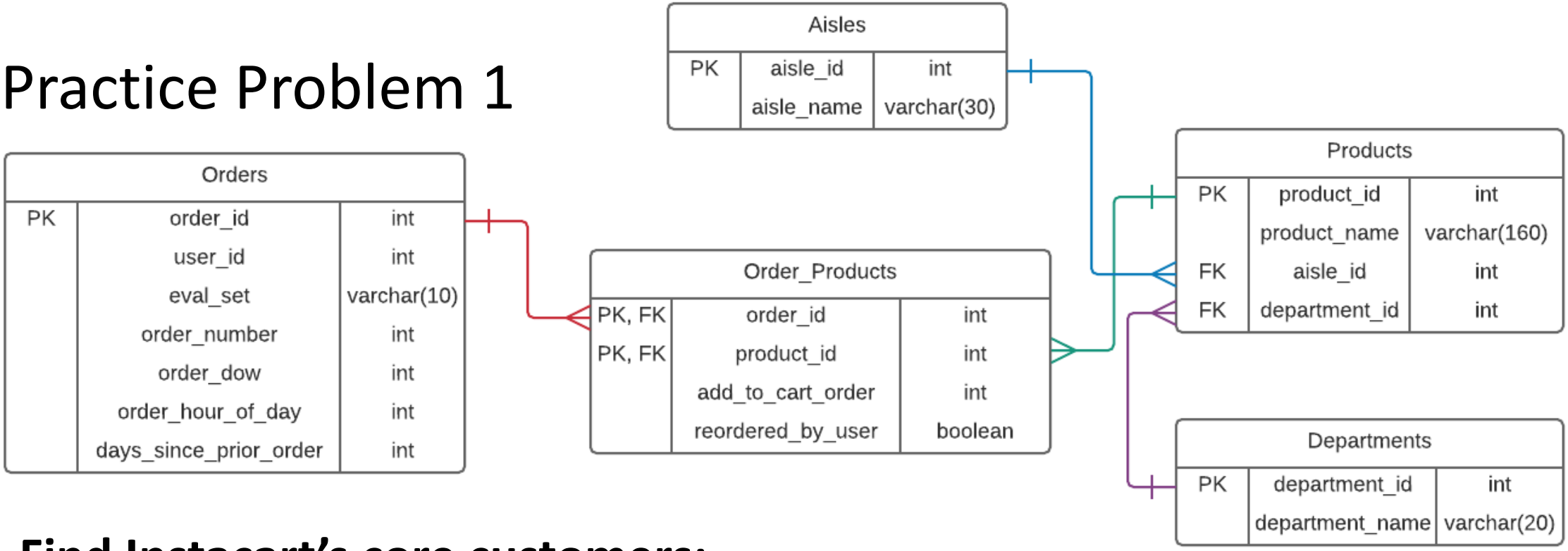
Employee

empid	firstname	lastname	salary	depid
1	Michael	Dell	100	5
2	Betty	Jennings	200	
3	Bill	Gates	0	5
4	Kay	McNulty	300	8
5	Jim	Gray	500	6
6	Gordon	Moore	400	6

Department

depid	depname
5	Executive
6	Research
7	Sales
8	Engineering

Practice Problem 1



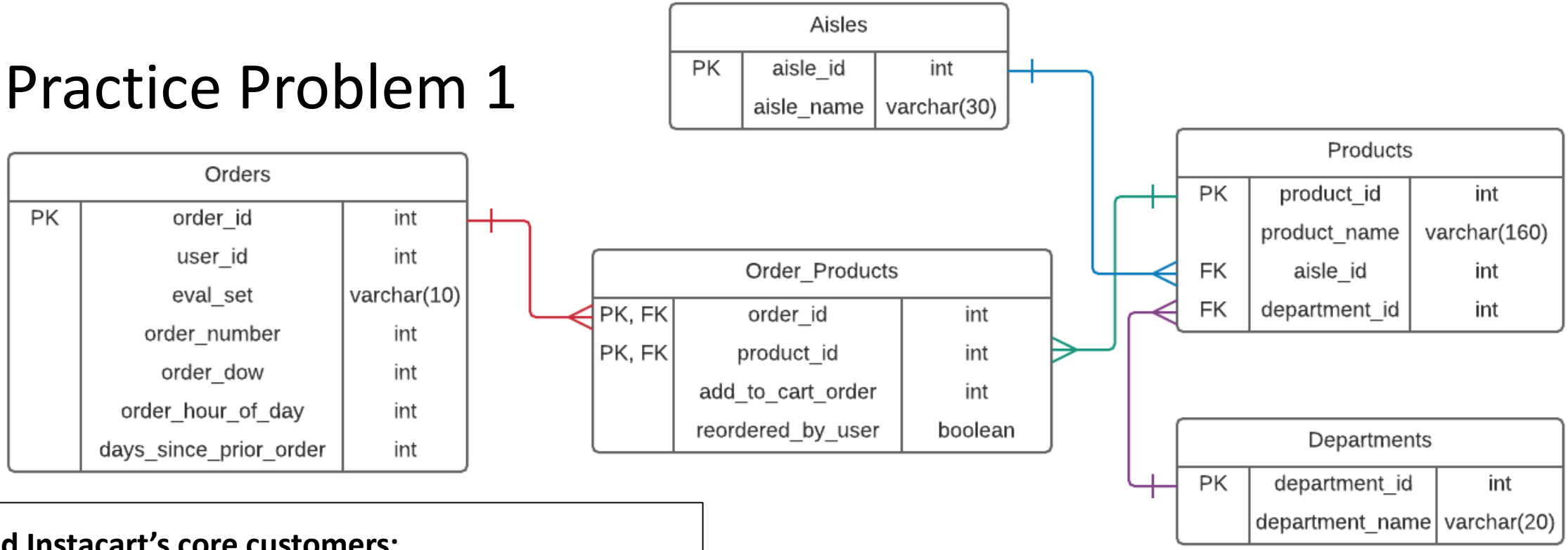
Find Instacart's core customers:

- customers who have placed at least 5 orders
- orders are no more than 7 days apart (use `days_since_prior_order`)
- orders contain at least 10 products (use `add_to_cart_order`)

Return `user_id` and number of orders placed by user.

Sort results by number of orders from highest to lowest.

Practice Problem 1



Find Instacart's core customers:

- customers who have placed at least 5 orders
- orders are no more than 7 days apart
- orders contain at least 10 products

Return user_id and number of orders placed by user.

Sort results by number of orders, from highest to lowest.

Aside from SELECT, FROM, and **ORDER BY, what clauses are needed to compute the answer?**

A) GROUP BY, HAVING, WHERE, JOIN

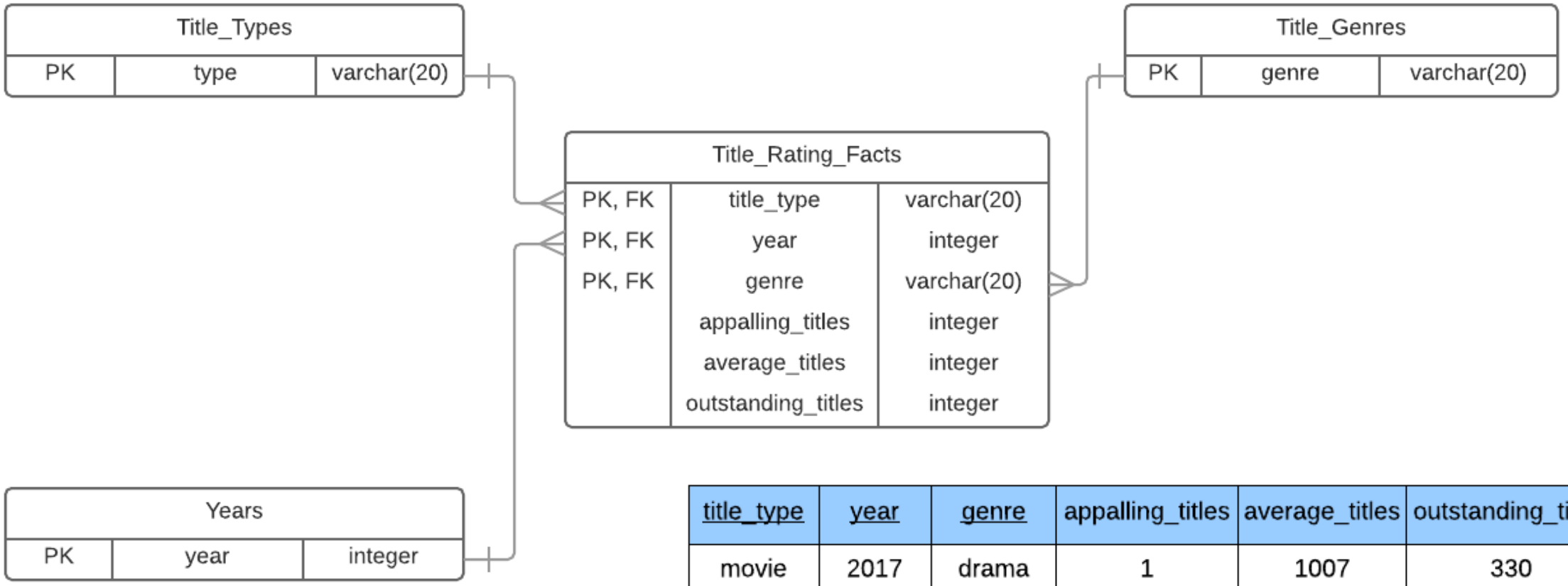
B) GROUP BY, HAVING, JOIN

C) GROUP BY, JOIN

D) GROUP BY, WHERE, JOIN

Solution: https://github.com/cs327e-fall2017/snippets/blob/master/instacart_aggregate_query_and_views.sql

Lab 2: Dimensional Schema



title_type	year	genre	appalling_titles	average_titles	outstanding_titles
movie	2017	drama	1	1007	330
movie	2017	comedy	3	633	144
movie	2017	horror	4	269	51
movie	2017	action	3	24	38
movie	2017	animation	1	64	19

Sample records

Notes:

- appalling titles: <= 2.0
- average titles: 2.1 - 7.9
- outstanding titles: >= 8.0