Joins - 9/13

Announcements

- Lab 1 will be released on Monday. You'll want to come to class with partners and laptops and sit next to your partners.
- Need to setup AWS and Lucidchart accounts.

Reading Quiz

- Q1: D.
- Q2: A. An `INNER JOIN` must have comparable attributes in order to be joined. Otherwise there is no point of an inner join.
- Q3: B. The join matches together rows with the same emp_dep and depid. If the condition does not match then the records with not appear in the result table (i.e. Sarah). If there are multiple matches however then all records with that match will appear in the result (i.e. Mike and Sunil).
- Q4: B. `ORDER BY` has a default of ascending order, but you can also specifiy descending.
- Q5: B. You can do `JOIN`s even on columns that are not primary keys.

Joins

Let's try to find what the primary key of the `order_products` table really is. We can already tell that it is not just the order_id itself.

To look at the table we can use `SELECT * FROM order_products LIMIT 30;` This will select some section of 30 records and return it as the result.

`SELECT count(*) FROM order_products;` Will return the count (the number of records) from the `order_products` table.

`SELECT count(*) FROM order_products WHERE order_id IS NOT NULL AND product_id IS NOT NULL;` Now we get the number of records where both fields are `NOT NULL`

`SELECT DISTINCT order_id, product_id FROM order_products;` This selects all the records with `DISTINCT` `order_id` and `product_id`s together.

`SELECT count(*) FROM (SELECT DISTINCT order_id, product_id FROM order_products);` Now we are running the previous query as a subquery so that we can get a count of the things we just listed out. Think of this as running a query on the result table of the `SELECT DISTINCT` query.

Practice Problem 1

There are 4 different FKs. FKs are in tables that "reference" other tables. So the FK should point to a PK in another table. Example the each record in product references an aisle. So the aisle_id in the products table points to one aisle_id in the aisle table.

Practice Problem 2

For a large order the add_to_cart_order must have a sum of over 100. So we find all the orders fitting this criteria by querying the `order_products` table and then we will be able to find all the `customer`s that are associated with these `order`s by futher joining with the `order` table and look at their user_id.

Give table aliases be putting the alias right after the table name. `FROM order o` Then you can reference orders using o.