

Tutorial 6: Basic Relational Algebra

Jin, Ziyang	Kim, Joon Hyung
# 34893140	# 35183128
f4a0b	11m8

February 2018

Consider a database consisting of the relations, with primary keys in bold:

customer(**cid**, cname, rating, salary)
order(**cid**, **iid**, **day**, qty)
item(**iid**, iname, type, price)

- 1 (a) Find the details (i.e., all attributes) of the customers who have a rating higher than 6 and earn less than \$125,000.

$$\sigma_{rating>6 \wedge salary<125000}(customer)$$

- 2 (b) Find the names and types of items which were ordered by some customer named 'Bob' who is rated above 5.

$$\pi_{iname,type}(\sigma_{cname='Bob' \wedge rating>5}(customer) \bowtie order \bowtie item)$$

- 3 (c) Consider the query $\pi_{iname,type}(item)$. Suppose item has 1000 tuples. Then how many tuples will the result of the above projection query contain? Explain your answer.

It depends on the actual instance of *item*. The number of tuples can vary from 1 to 1000.

(iname type) is not a key. There is also no uniqueness constraint on iname and type. So it is possible that iname and type are the same for these 1000 tuples. In this case, the resulting projection will remove duplicates and result in 1 tuple. It is also possible that iname and type are distinct from tuple to

tuple, so the resulting projection will have 1000 distinct tuples. Therefore, it is possible that some of the (**iname type**) pairs are duplicates and some of them are distinct so the result can vary from 1 to 1000.

- 4 (d) Suppose in addition to the key dependencies identified by the primary keys in the given relations, we are told that every item name uniquely determines its type, i.e., that the FD **iname** \rightarrow **type** holds. Suppose there are 950 distinct item names in the item table.

1. Then how many tuples will be present in the result of the following RA^{SPJ} query. Explain your answer.

$$\pi_{iname,type}(item)$$

There will be exactly 950 tuples in the result. Because there are 950 distinct item names in the item table. If we project (**iname type**) pair, none of the resulting pairs will be removed as a duplicate because each pair has a unique **iname**. So there will be 950 tuples in the result.

2. Would your answer change if the query instead was:

$$\pi_{iname,type,price}(item)$$

Yes. There will be at least 950 tuples in the result.

The resulting tuple is removed only if all of the attributes in the tuple are the same as another. Since **iname**'s are distinct, the entire resulting tuples are distinct. Therefore, there is no duplicate tuple to be removed in the result. The result cannot have fewer than 950 tuples.

However, **iname** is not a key, nor is it unique in the entire table. So it is possible that there are 2 or more tuples with different **iid**, different **price**, but same **iname** and same **type**. In the result, because the **price** is different, the tuples with same **iname**, **type** but different **price** are not duplicates so they will not be removed. So there can be more than 950 tuples.

- 5 (e) Find the type and name of those items, whose price is at most \$100 OR which were ordered by the customer with **cid**=123.

$$\pi_{type,iname}(\sigma_{price \leq 100 \vee (item.iid=order.iid \wedge cid=123)}(item \bowtie_{true} order))$$

- 6 (f) Consider the query: "Find the names of customers who did not order any laptop, i.e., did not order any item of type laptop" and the RA^{SPJ} query. Does the RA^{SPJ} query above correctly capture the English query above? Justify your answer.

$$\pi_{cname}(customer \bowtie order \bowtie \sigma_{type \neq 'laptop'}(item))$$

No.

Reason 1: Edge Case

The join between customer and order only selects customers that have ordered something. If a customer has never ordered anything, the customer still qualifies the case "who did not order any laptop". However, in this join, these qualifying customers who have not ordered anything are lost. Therefore, the RA^{SPJ} query above does not correctly capture the English query above.

Reason 2: Semantics of NOT

Consider the case where a customer has ordered a laptop and another item that is not laptop. In the English query, we do not want the customer to be included in the result because the customer has ordered a laptop. In the RA^{SPJ} query, the query only eliminates the 'laptop' items and let go other types of items. So the customer who ordered a laptop and another item will still be selected — by the tuple of "another" item the customer ordered.