2024.10.21 7:42 ~ 9:30

(a) You are given a relation $R(A, B, C, D)$. Identify all sets of attributes which could be keys.

A	B	C	D
1	1	2	1
1	3	4	1
2	1	4	4
2	3	2	1
2	4	3	2
3	2	1	2
3	4	2	1
4	1	3	4
4	3	3	2
4	4	1	4

$\binom{4}{1}$ • Neither of a single attribute can uniquely identify the tuples since there are 4 values per attribute and 10 tuples

$\binom{4}{2}$ • $\{A, B\}$ is a key

$\{B, C\}$ is a key

$\{C, D\}$ is not a key, 3 tuples have $C = 2$ and $D = 1$

$\{A, C\}$ is not. 2 tuples have $A = 4$ and $C = 3$

$\{A, D\}$ is not $\Rightarrow A = 1, D = 1$

$\{B, D\}$ is not $\Rightarrow B = 3, D = 1$

$\binom{4}{3}$ • Since $\{A, B\}$, $\{B, C\}$ are keys, exclude $\{A, B, C\}$, $\{A, B, D\}$ and $\{B, C, D\}$ since they are not minimal set for identification

$\{A, C, D\}$ is a key

$\binom{4}{4}$ • no need to check $\{A, B, C, D\}$ since $\{A, B\}$. $\{B, C\}$ are keys

(b) Assuming the following attribute names with the same values as before, which keys would still be a reasonable choice for a production database given the new domain knowledge? Consider that more rows could be added in the future. The attribute `user_order_id` can be assumed to be unique among orders by the *same* user.

Orders

user_id	user_order_id	amount	product_id
1	1	2	1
1	3	4	1
2	1	4	4
2	3	2	1
2	4	3	2
3	2	1	2
3	4	2	1
4	1	3	4
4	3	3	2
4	4	1	4

{user_id, user_order_id} is a reasonable choice, Since the order id given a user is unique, the pair can identify any row in the table.

Exercise 2 *Relational Algebra* (5 Points)

Given the following relational model of a supermarket chain where underlined attributes indicate primary keys and overlined attributes indicate foreign keys:

product($\underline{\texttt{id}}$, price, name)

store($\underline{\texttt{id}}$, city)

customer($\underline{\texttt{id}}$, firstname, lastname)

sold_in($\underline{\overline{\texttt{product\_id}}}$, $\underline{\overline{\texttt{store\_id}}}$) with foreign keys `product_id` referencing the attribute `id` of `product` and `store_id` referencing the `id` of `store`

order($\underline{\texttt{id}}$, $\overline{\texttt{customer\_id}}$, $\overline{\texttt{product\_id}}$, amount) with foreign keys `customer_id` referencing `id` of `customer` and `product_id` referencing `id` of `product`

Formulate the following queries using relational algebra. In this lecture, only the following operators are allowed: $\pi, \sigma, \bowtie, \rho (\equiv \beta), \leftarrow, \wedge, \vee, =, \leq, \geq, \times, \neq, \neg, -, \div$

(a) Find all cities where a store is present.

projection removes duplicates. "if something is true, then saying it twice won't make it any truer"

Answer: $\pi_{city}(store)$

(b) Find all stores in Bonn and Berlin. If not otherwise stated, the result should have all attributes of the specific relation, i.e. both `id` and `city` in this case.

Answer: $\sigma_{city = 'Bonn' \vee city = 'Berlin'}(store)$

(c) Find all names of products sold in stores in Bochum.

Answer:

$$\pi_{name}(\underbrace{(sold\_in \bowtie (\rho_{store\_id \leftarrow id}(\sigma_{city='Bochum'}(store)))) \bowtie (\rho_{product\_id \leftarrow id}(product))}_{\text{Schema: \{product\_id, store\_id, city, price, name\}}})$$

no pizza : (oder join product join customer)

(d) Which customers never ordered pizza?

$$\pi_{customer\_id,\, firstname,\, lastname}\left(\sigma_{name \neq 'pizza'}\left((order \bowtie (\rho_{product\_id \leftarrow id}(product))) \bowtie (\rho_{customer\_id \leftarrow id}(customer)) \right)\right)$$

(e) Which customers bought all the products?

$$\pi_{customer\_id,\, firstname,\, lastname}\left((\rho_{customer\_id \leftarrow id}(customer)) \bowtie ((\pi_{customer\_id,\, product\_id}(order)) \div (\rho_{product\_id \leftarrow id}(\pi_{id}(product)))) \right)$$