FIT9132 Introduction to Databases

Week 4 Applied Class Suggested Solution

FIT Database Teaching Team

FIT9132 2022 S1

FIT9132 Introduction to Databases

Author: FIT Database Teaching Team

4.1 The Relational Model

4.1.1 Class Discussion

Given these two relations:

CUSTOMER (cust_id, cust_name, cust_address)

ORDER (order_id, order_date, cust_id)

Assuming that a customer may have any number of orders and each order is placed by a single customer, discuss the following terms based on the above relations:

1. Relation: *A named set of attributes, consisting of a heading and a body. The heading is the schema, the body is the set of tuples (the state at a point in time)*
2. Attribute: *A characteristic of an object/entity that we wish to record eg. cust_name*
3. Domain: *A set of atomic (indivisible) values from which an attribute's values are drawn. Consists of a name, data type and data format eg. address domain: 30 character string containing street, suburb and postcode*
4. Tuple: *A set of related attribute values describing a particular instance of the relation – in file terminology called a row*
5. Degree and cardinality of a relation:

   Degree: number of attributes in a relation;

   Cardinality: number of tuples
6. Primary key and foreign key:

   Primary key: an attribute or minimal set of attributes which uniquely identify each tuple; The primary key of CUSTOMER is the cust_id, and the primary key of ORDER is the order_id.

   Foreign key: an attribute(s) in a relation that exists in the same or another relation as Primary Key; The foreign key is the cust_id in the ORDER relation.

4.1.2 Choosing the Primary key

1. In any relation, tuples must be unique. However, in many cases, the set of all the attributes in a relation is not considered a candidate key. Why not?

   Although all of the attributes in a relation is a superkey, the candidate key is selected on the basis of a ***minimal*** *superkey. For a given relation there is usually a smaller set of attributes that provide a superkey so there is no need to consider the full set of attributes as a starting point.*

   On the other hand, suppose we do have a relation where the set of all attributes is a candidate key. In this case, show that this set must, therefore, be the only candidate key and hence the primary key.

   If the set of all attributes is a candidate key (ie. a minimal superkey), there can be no other superkeys and hence it must be the only candidate key and the primary key.

2. Underlined attribute(s) is the primary key

    ORDER (order_id, order_date, cust_id)

    ORDERLINE (order_id, prod_no, ol_qtyordered, ol_lineprice)

    PRODUCT (prod_no, prod_desc, prod_unitprice)

    The foreign keys are order_id and prod_no in ORDERLINE. order_id in ORDERLINE refers to order_id in ORDER and prod_no in ORDERLINE refers to prod_no in PRODUCT.

3. Consider a relation that depicts a dental surgery appointment system.
    APPOINTMENT (dentist_id, dentist_name, patient_id, patient_name, appointment_datetime, surgeryroom_no)

    Candidate keys:
    (dentist_id, appointment_datetime)
    (patient_id, appointment_datetime)
    (surgeryroom_no, appointment_datetime)
    Primary key:
    One of the candidate keys

4.2 Relational Algebra

4.2.1 Relational Algebra Exercise

HOTEL (hotel_no, hotel_name, hotel_city)
ROOM (room_no, hotel_no, room_type, room_price)
BOOKING (hotel_no, guest_no, bdate_from, bdate_to, room_no)
GUEST (guest_no, guest_name, guest_address)

1. List the names and cities of all hotels

    $R = \pi_{\text{hotel\_name, hotel\_city}}\ \text{HOTEL}$

2. List all single rooms with a price below $50

    $R = \sigma_{\text{room\_type='single' and room\_price < 50}}\ \text{ROOM}$

3. List the names of all hotels in Melbourne

    $R = \pi_{\text{hotel\_name}}(\sigma_{\text{hotel\_city='Melbourne'}}\ \text{HOTEL}))$

4. List all names of hotels which have presidential suite room

    $\text{PSUITE\_NO} = \pi_{\text{hotel\_no}}\ (\sigma_{\text{room\_type = 'presidential suite'}}\ \text{ROOM})$

    $R = \pi_{\text{hotel\_name}}\ (\text{PSUITE\_NO} \bowtie (\pi_{\text{hotel\_no, hotel\_name}}\ \text{HOTEL}))$

or

$R = \pi_{hotel\_name} ((\pi_{hotel\_no} (\sigma_{room\_type = 'presidential\ suite'} ROOM)) \bowtie$

$(\pi_{hotel\_no,\ hotel\_name} HOTEL))$

5. List the price and type of all rooms at the Grosvenor Hotel

$GROSVENOR\_NO = \pi_{hotel\_no} (\sigma_{hotel\_name = 'Grosvenor'} HOTEL)$

$R = \pi_{room\_price,\ room\_type} (GROSVENOR\_NO \bowtie (\pi_{hotel\_no,\ room\_price,\ room\_type} ROOM))$

or

$R = \pi_{room\_price,\ room\_type} ((\pi_{hotel\_no} (\sigma_{hotel\_name = 'Grosvenor'} HOTEL)) \bowtie$

$(\pi_{hotel\_no,\ room\_price,\ room\_type} ROOM))$

6. List all names and addresses of guests currently staying in deluxe room of any hotel (assume that if the guest has a tuple in the BOOKING relation, then they are currently staying in the hotel)

$DELUXE\_NO = \pi_{hotel\_no,\ room\_no} (\sigma_{room\_type = 'deluxe'} ROOM)$

$GUEST\_DELUXE\_NO = \pi_{guest\_no} (DELUXE\_NO \bowtie (\pi_{guest\_no,\ hotel\_no,\ room\_no} BOOKING))$

$R = \pi_{guest\_name,\ guest\_address} (GUEST\_DELUXE\_NO \bowtie GUEST)$

or

$R = \pi_{guest\_name,\ guest\_address} ((\pi_{guest\_no} ((\pi_{hotel\_no,\ room\_no} (\sigma_{room\_type = 'deluxe'} ROOM)) \bowtie$

$(\pi_{guest\_no,\ hotel\_no,\ room\_no} BOOKING))) \bowtie GUEST)$

7. List all names and addresses of guests currently staying at the Grosvenor Hotel (assume that if the guest has a tuple in the BOOKING relation, then they are currently staying in the hotel)

$GROSVENOR\_NO = \pi_{hotel\_no} (\sigma_{hotel\_name = 'Grosvenor'} HOTEL)$

$GROSVENOR\_BOOKING = \pi_{guest\_no} (GROSVENOR\_NO \bowtie (\pi_{guest\_no,\ hotel\_no} BOOKING))$

$R = \pi_{guestl\_name,\ guest\_address} (GROSVENOR\_BOOKING \bowtie GUEST)$

or

$R = \pi_{guest\_name,\ guess\_address} ((\pi_{guest\_no} ((\pi_{hotel\_no} (\sigma_{hotel\_name = 'Grosvenor'} HOTEL)) \bowtie$

$(\pi_{guest\_no,\ hotel\_no} BOOKING))) \bowtie GUEST)$

4.2.2 Additional Relational Algebra Exercise

CUSTOMER (cust_id, cust_name, cust_address)

PRODUCT (prod_id, prod_desc, prod_unitprice, prod_stock)

STAFF(staff_name, staff_position)

SALE (cust_id, sale_date, prod_id, sale_qty, sold_by)
*Note that sold_by value is the name of staff who made the sale

1. List names of customers and descriptions of products bought by the customers. How many tuples will be returned by the relational algebra query that you have constructed as your answer?

   $R1 = \pi_{cust\_id,\ cust\_name}\ CUSTOMER$

   $R2 = \pi_{cust\_name,\ prod\_id}((\pi_{cust\_id,\ prod\_id}\ SALE) \bowtie R1)$

   $R = \pi_{cust\_name,\ prod\_desc}((\pi_{prod\_id,\ prod\_desc}\ PRODUCT) \bowtie R2)$

   → 5 tuples

2. List all names which are shared by customers and staff.

   $R1 = \pi_{cust\_name}\ CUSTOMER$

   $R2 = \pi_{staff\_name}\ STAFF$

   $R = R1 \cap R2$

3. List descriptions of products that haven't been sold

   $R = \pi_{prod\_desc}(((\pi_{prod\_id}\ PRODUCT) - (\pi_{prod\_id}\ SALE)) \bowtie (\pi_{prod\_id,\ prod\_desc}\ PRODUCT))$

4. List names of clerks who don't have any sales yet

   $R = (\pi_{staff\_name}(\sigma_{staff\_position="Clerk"}\ STAFF)) - (\pi_{sold\_by}\ SALE)$

5. List positions of staff who have made sales

   $R = \pi_{staff\_position}((\pi_{sold\_by} SALE) \bowtie_{(staff\_name=sold\_by)} STAFF)$