4.1.1 Class Discussion

Given these two relations:
CUSTOMER (cust_id, cust_name, cust_address)
ORDER (order_id, order_date, cust_id)

CUST_ID	CUST_NAME	CUST_ADDRESS	CUST_GENDER
1	Jack	6 Jalan Jaya 2	M
2	Jill	185 Jalan Ampang	F
3	John	16-02, Jalan Permas 9/2	M

ORDER_ID	ORDER_DATE	CUST_ID
1	5/3/2021	1
2	5/4/2021	1
3	6/4/2021	1
4	5/4/2021	2
5	6/4/2021	3
6	8/4/2021	3

1. Identify the following terms based on the above CUSTOMER and ORDER relations:

a) Relation - customer and order

b) Attribute - cust_id, 4 for customer, 3 for order

c) Domain - CUST_NAME 20 character string

d) Tuple {1, Jack, 6 Jalan Jaya 2, M}, 6, 3 tuples

e) Degree of a relation and - 4,3

f) Cardinality - 3 for CUSTOMER, 6 for ORDER

g) Primary key - CUST_ID for CUSTOMER and ORDER_ID for ORDER

h) Foreign key - no foreign key for CUSTOMER and CUST_ID for ORDER

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?**

Table: Employee

Emp_SSN	Emp_Number	Emp_FName	Emp_Lname
---------	----------	----------	----------
123456789	226	Steve	John
999999321	227	Steve	Smith
888997212	228	Sherry	Smith
777778888	229	Robert	John

Super Key:

Emp_SSN

Emp_Number

Emp_SSN, Emp_Number, Emp_FName, Emp_Lname

Emp_SSN, Emp_Number

Emp_SSN, Emp_Number, Emp_FName

Emp_FName, Emp_Lname

Candidate key:

Emp_SSN,

Emp_Number

Emp_FName, Emp_Lname

Primary Key:

- **Emp_SSN**
- Emp_no

2. 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.

PET ID	VISIT DATE	PROCEDURE
246ROVER	JAN 13/2019	01 RABIES VACCINATION
246ROVER	MAR 27/2019	10 EXAMINE AND TREAT WOUND
246ROVER	APR 02/2019	05 HEART WORM TEST
298SPOT	JAN 21/2019	08 TETANUS VACCINATION
298SPOT	MAR 10/2019	05 HEART WORM TEST
341MORRIS	JAN 23/2019	01 RABIES VACCINATION
341MORRIS	JAN 13/2019	01 RABIES VACCINATION
519TWEEDY	APR 30/2019	20 ANNUAL CHECK UP
519TWEEDY	APR 30/2019	12 EYE WASH

Superkey:

- PET ID, VISIT DATE, PROCEDURE

Candidate Key:

PET ID, VISIT DATE, PROCEDURE

Primary Key:

PET ID, VISIT DATE, PROCEDURE

3. **Identify the primary key and foreign key for these three relations:**

ORDER (ORDER_ID, ORDER_DATE, CUST_ID)

ORDERLINE (ORDER_ID, PROD_NO, OL_QTYORDERED, OL_LINEPRICE)

PRODUCT (PROD_NO, PROD_DESC, PROD_UNITPRICE)

ORDER

PK: ORDER_ID

FK: CUST_ID

ORDERLINE

PK: ORDER_ID and PROD_NO together

FK: ORDER_ID, PROD_NO

PRODUCT

PK : PROD_NO

FK : None

Relationships:

Order - Product (Many-to-many)

Order - Orderline (one to many)

Product - Orderline (one to many)

4. Identify the superkey(s), candidate key(s) and the primary key for the relation if the following business rules are applicable:

- A dentist can only see a single patient at a particular date and time
- A dentist treats a patient in a particular surgery room, and
- A patient can see the same dentist multiple times

APPOINTMENT

dentist_id	dentist_name	patient_id	patient_name	appointment_datetime	surgery_roomno
D1	Jack	P1	Jill	5/4/2020 10am	1
D1	Jack	P2	Smith	5/4/2020 12pm	1
D2	John	P3	Mary	5/4/2020 10pm	2
D2	John	P3	Mary	6/4/2020 2pm	2
D3	Will	P2	Smith	6/4/2020 4pm	1
D3	Will	P4	Doe	7/4/2020 6pm	3

Superkey:

dentist_id, dentist_name, appointment_datetime, surgery_room

Dentist_id, patient_id, appointment_datetime

dentist_id, appointment_datetime, surgery_roomno

dentist_id, appointment_datetime

Dentist_id, dentist_name, patient_id, patient_name, appointment_datetime

Dentist_id, dentist_name, patient_id, patient_name, appointment_datetime, surgery_room

patient_id, appointment_datetime

Candidate key:

~~dentist_id, patient_id, appointment_datetime?~~

dentist_id, appointment_datetime

appointment_datetime, surgery_roomno

patient_id, appointment_datetime

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)

π σ ⋈

1. List the names and cities of all hotels

π(HOTEL_NAME, HOTEL_CITY)(HOTEL)

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

σ ROOM_TYPE = 'SINGLE' and ROOM_PRICE < $50 (ROOM)

3. List the names of all hotels in Melbourne

π(HOTEL_NAME)(σ HOTEL_CITY="Melbourne"(HOTEL))

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

π(HOTEL_NAME)(σ ROOM_TYPE = 'presidential suite'(HOTEL))

S5. List the price and type of all rooms at the Grosvenor Hotel

π (ROOM_TYPE, ROOM_PRICE)σ(HOTEL_NAME = 'Grosvenor Hotel'(HOTEL ⋈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)

πGUEST_NAME, GUEST_ADDRESS(σROOM_TYPE = deluxe room(ROOM ⋈ GUEST ⋈ BOOKING))

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)

π(names, address)(Booking⋈(Guest number = guest number)Guest), using right outer join

4.2.2. Advanced Relational Algebra Exercise

Considers these four relations:

CUSTOMER (cust_id, cust_name, cust_address)

PRODUCT (prod_no, prod_desc, prod_unitprice, prod_stock)

STAFF(staff_name, staff_position)

SALE (cust_id, sale_date, prod_no, sale_qty, sold_by)

***Note that sold_by value is the staff who made the sale**

π σ ⋈

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

P1 = π(prod_no, prod_desc)(PRODUCT) #retrieve all prod_desc

C1 = π(cust_id, cust_name)(CUSTOMER) #retrieve all cust_name

S1 = π(cust_id, prod_no) #filter out unnecessary attributes from SALE

SP = P1 ⋈ S1 ⋈ C1 #add prod_desc and cust_name to SALE

A1 = π(cust_name, prod_desc)(SPC) #project the cust_name and prod_desc for each tuple in sale

5 tuples will be returned

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

N1 = π(cust_name) (CUSTOMER)

N2 = π(staff_name) (STAFF)

N3 = N1 ∩ N2

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

- S1 = π(prod_no)(SALE)
- P1 = π(prod_no)(PRODUCT)
- PD1 = π(prod_no,prod_desc)(PRODUCT)
- NS1 = P1-S1
- π(prod_desc)(NS1 ⋈ PD1)

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

$$\pi(\text{staff\_name})(\sigma_{\text{staff\_position=clerk AND sold\_by=null}}(\text{SALE} \rightarrowtail \text{STAFF}))$$

a = π (staff_name (STAFF) ⋊ sold_by (SALE))

out = π a (σ staff_position = clerk and sale_qty= 0)

5. List categories (positions) of staff who have made sales