officer = data_officer in cpdb.

Q1: Π first_name, middle_initial, last_name (σ join_date $\geq$ March 15, 2020 (officer))

Q2: The numbers of complaint reports filled by officers require aggregate functions (Counting), and can not be expressed in relational algebra.

Q3: find MAX(appointed_date):

$R1 := \sigma O1.appointed\_date \leq O2.appointed\_date$ (data_officer X data_officer)

Select officer with no later appointed_date:

$R2 := \sigma O1.appointed\_date = O2.appointed\_date$ (R1)

Project the required fields:

π O.first_name, O.middle_initial, O.last_name(R2)

O = data_officer table

Q4: ~~+ Self-Join for 2 distinct complaints~~

Join data_officer, data_officer allegation, data_allegation:

R1 := data_officer $\bowtie$ O.id = OA.officer_id((data_officerallegation) $\bowtie$ A.crid = OA.allegation_id (data_allegation)

Find distinct officer Pairs:

R2 := π OA.officer_id, A.crid(R1)

Self-Join 2 times to find distinct complaints:

1. R3 := σ R2.officer_id = R2'officer_id $\wedge$ R2.crid $\neq$ R2'.crid (R2 X R2')
2. R4 := σ R3.officer_id = R3'.officer_id $\wedge$ R3.crid $\neq$ R3'.crid(R3 X R3')

(continue)

Q4: Join Back with data_officer:

$R5 := data\_officer \bowtie O.id = R4.officer\_id\ R4$

Projection:

$\pi\ O.id, O.first\_name, O.last\_name\ (R5)$

Q5: Counting numbers of officers, and numbers of population in each district will require aggregate function and can not be described in relation algebra.

Q6: self join data_officerallegation:

$R1 := \sigma\ OA1.allegation\_id = OA2.allegation\_id \wedge OA1.officer\_id < OA2.officer\_id\ (data\_officerallegation \times data\_officerallegation)$

Join with data_officer → last_unit_id

officer 1: $R2 := R1 \bowtie OA1.officer\_id = O1.id\ (data\_officer)$

officer 2: $R3 := R2 \bowtie OA2.officer\_id = O2.id\ (data\_officer)$

Select officer from different PoliceUnit:

$R4 := \sigma O1.last\_unit\_id \neq O2.last\_unit\_id\ (R3)$

Projection

$\pi\ least(OA1.officer\_id, OA2.officer\_id), O1.last\_unit, greatest$ as officer1_id, O1.last_unit_id as officer1_unit, greatest(OA2.officer_id, OA2.officer id) as officer2_id, O2.last.unit_id as officer2_unit (R4)

Q7: The average is a aggregate function, and can not be calculated, or expressed using relational algebra.

Q8: self join to compare sustained Counts.:

$R1 := \sigma O_1.rank = O_2.rank \wedge O_1.sustained\_count >$
$O_2.sustained\_count$ (data_officer X data_officer)

Select officer with maximum sustained count

$R2 := \sigma O_1.rank = O_2.rank \wedge O_1.sustained\_count > O_2.sustained\_count$ (data officer X data_officer)

subtract to find officers with maximum sustained count:

$R3 := \pi O_1.rank, O_1.sustained\_count(R_1) - \pi O_2.rank, O_2.sustained\_count(R_2)$

Projection

$R4 := \pi O.rank, O.sustained\_count(R_3)$

Q9: Counting allegations with "SU" or "EX" require aggregate function (Count) and can not be expressed in relational algebra

Q10: Counting Complaints require aggregate function (COUNT) and can not be expressed in relational algebra. Grouping based on categories, allegation, name, is also not expressable in relational algebra.