Warmup 1

1.

$$\{t \mid \exists s \in Tree(s[t\_id] = t[t\_id] \wedge \\ \forall r \in Tree(r[years\_to_maturity] \leqslant s[years\_to_maturity]))\}$$

2.

$$\{n \mid \exists n_1 \in Nursery(n_1[n\_id] = n[n\_id] \wedge \\ \exists s \in Sells(n_1[n\_id] = s[n\_id] \wedge \\ \forall t \in Tree(t[t\_id] = s[t\_id] \wedge t[height] \geqslant 10)))\}$$

3.

$$\{n \mid \exists n_1 \in Nursery(n_1[n\_id] = n[n\_id] \wedge n_1[state] =' California' \wedge \\ \exists s \in Sells(n_1[n\_id] = s[n\_id] \wedge s[price] \geqslant 20 \wedge \\ \exists t \in Tree(t[t\_id] = s[t\_id] \wedge t[height] = 10)))\}$$

4.

$$\{n \mid \forall n_1 \in Nursery(n_1[n\_id] = n[n\_id1] \wedge \\ \forall n_2 \in Nursery(n_2[n\_id] = n[n\_id2] \wedge \\ \exists s_1 \in Sell(s_1[n\_id] = n_1[n\_id] \wedge \\ \exists s_2 \in Sell(s_2[n\_id] = n_2[n\_id] \wedge \\ n_1[price] > n_2[price])))))\}$$

5.
(a)Find id and name of trees that sold by the nursery named as "Johny Appleased"
(b)Find id and name of all Nurseries that sold any kind of trees for highest price

Warmup 2

1.
A query is safe when the result is a finite set of entities.
2.
$\{t \mid \neg(t \in Tress) \wedge t \in Shrubs\}$ is safe, since the result is finite.
$\{t \mid \neg(t \in Tress) \vee t \in Shrubs\}$ is not safe, because the result can be infinite.

Warmup 3

1.
Person (p_id, (first_name, middle_name, last_name), age, (street_name, state, zip_code, country), {phone_number})
Gardener (g_id, years_of_experience)
Tree (species, max_height, years_to_maturity)
2.
Job: relating gardener with person
Cultivate: relating gardener with tree
3.

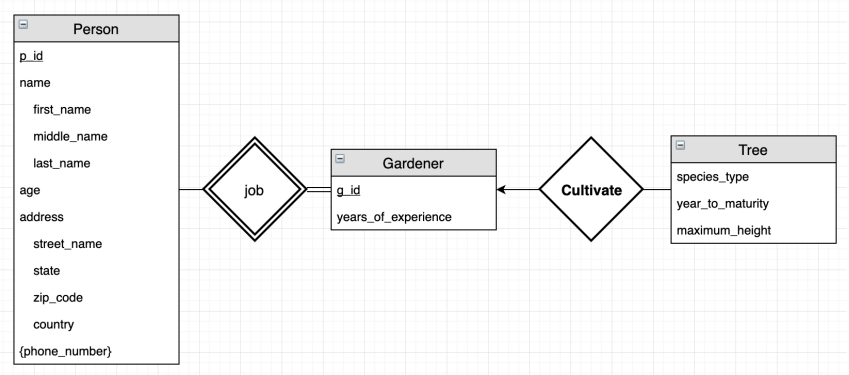

Figure 1: ER diagram

4.

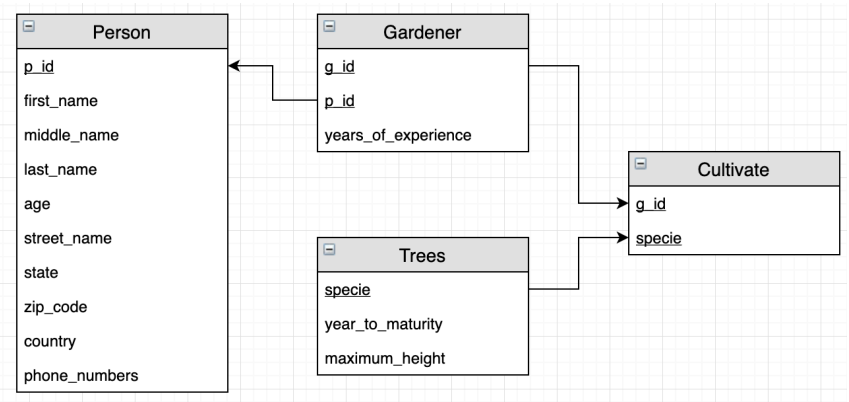

Figure 2: Database schema

Problem 4

1.
User (user_id, (first_name, last_name), email, minor, (guardian_first_name, guardian_last_name))
Explanation : The customer and passenger can be the same person, and the information for customer overlapped with the information for passenger. Therefore, I create a new entity called user to combine customer and passenger.

Route(route_id, stop_number, origin, destination, stop1, stop2)
Explanation : The entity route indicates all possible routes. A route_id is used to identify a route. The stop_number is derived attribute to show the number of stops in the current route.

Flight(flight_number, departure_time, route_id, arrive_time, aircraft_id,
first_section_available_number, {first_section_available_seats},
business_section_available_number,{business_section_available_seats},
economy_section_available_number,{economy_section_available_seats})
Explanation : The entity flight saves the information of a fight. The reason why I didn't create a entity for aircraft is that the same aircraft may serve different flight and it's impossible to record available seats for different flight in one entity.
2.

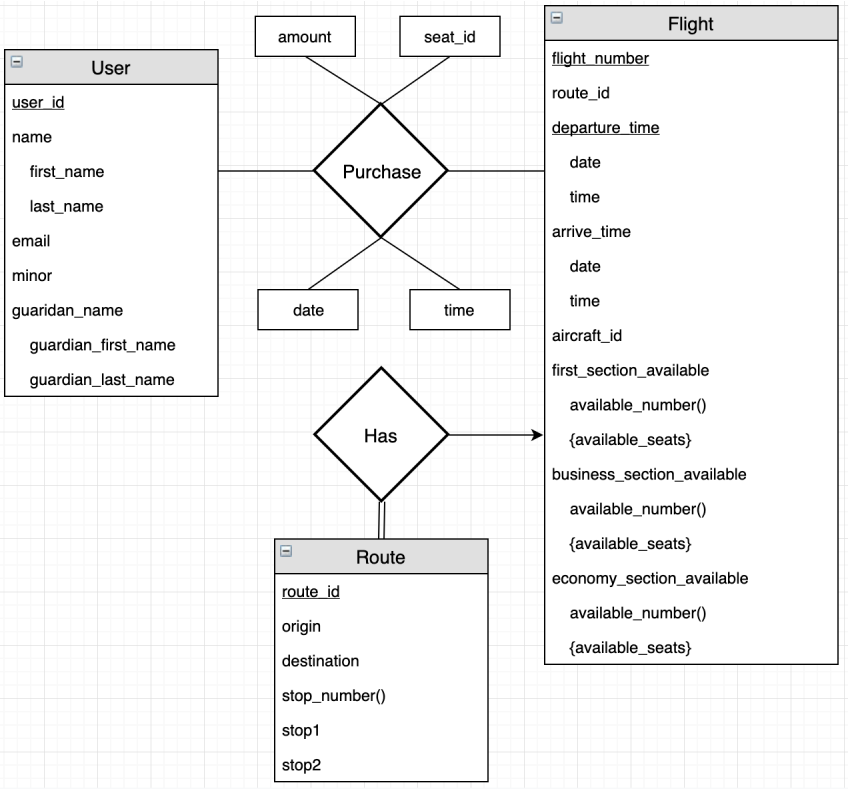

Figure 3: ER diagram

3.

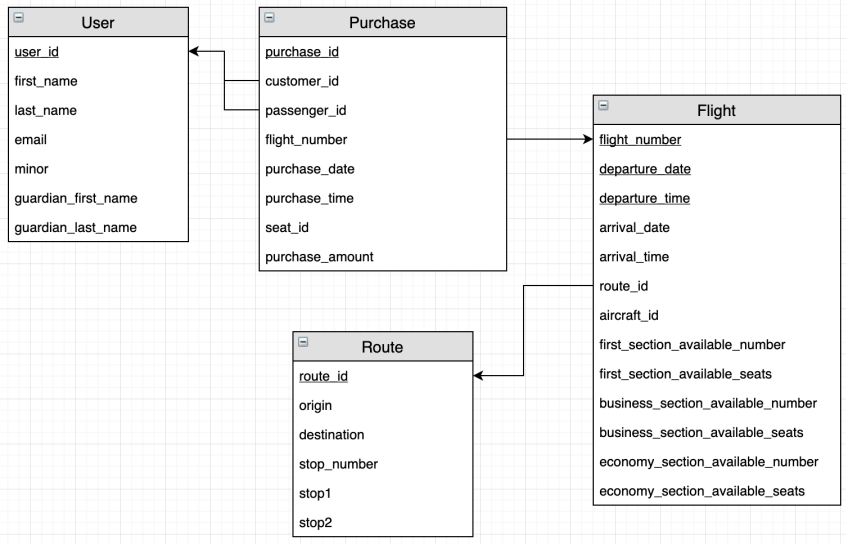

Figure 4: ER diagram

Problem 5

1.
Relational algebra
select the route_id from the Table Route where origin is "PVD" and the destination is "LAX" and the number of stops is no more than 1

$$Temp1 \leftarrow \pi_{route\_id}(\sigma_{origin='PVD' \wedge destination='LAX' \wedge stop\_number \leqslant 1}(Route))$$

natural join Temp1 and Flight and then select the flight that the departure date is September 30, 2019 and has business seats available

$$Result \leftarrow \sigma_{departure\_date='September30,2019' \wedge business\_section\_available\_number>0}(Flight \bowtie Temp1)$$

Tuple relational calculus

$$\{f \mid f \in Flight \wedge f[departure\_date] = "Sep30, 2019" \wedge f[business\_section\_available\_number] > 0 \wedge \exists r \in Route(r[route\_id] = f[route\_id] \wedge r[origin] = "PVD" \wedge r[destination] = "LAX" \wedge r[stop\_number] \leqslant 1)\}$$

2.
Let "destination" be the group's destination.
Let "location" be their current airport location.
Let "date" be the current date.
Let "time" be the current time.
Relational algebra
select the route_id from the Table Route where origin is current location and destination is their original destination

$$Temp1 \leftarrow \pi_{route\_id}(\sigma_{origin='location' \wedge destination='destination' \wedge stop\_number \leqslant 1}(Route))$$

natural join Temp1 and Flight and then select the flight that the departure date is later than current time

$$Temp2 \leftarrow \sigma_{departure\_date \leqslant 'date' \wedge departure\_time>'time'>0}(Flight \bowtie Temp1)$$

select the flights that have more than 3 available seats

$$Result \leftarrow \sigma_{first\_section\_available\_number+business\_section\_available\_number+economy\_section\_available\_number \geqslant 3}(Temp2)$$

Tuple relational calculus

$$\{f \mid f \in Flight \wedge f[departure\_time] > "time" \wedge f[departure\_date] \geqslant "date" \wedge f[first\_section\_available\_number] + f[business\_section\_available\_number] + f[economy\_section\_available\_number] \geqslant 3 \wedge \exists r \in Route(r[route\_id] = f[route\_id] \wedge r[origin] = "location" \wedge r[destination] = "destination")\}$$

Problem 6

1.

- prompt response: TAs give prompt response on Piazza.
- great recitation: Recitation is even better than lecture.
- too much homework: I spent more than 10 hours on both HW1 and HW2.

2."Big Data" processing
3. Capture for recitation.