**Database Assignment 2  
k21-3906  
  
Question 1  
  
1.Find the colors of boats reserved by Albert.**π(COLOR)(σ(SNAME = 'Albert')(SAILORS ⨝ RESERVES ⨝ BOATS))  
  
**2.Find all sailor IDs of sailors who have a rating of at least 8 or reserved boat 103.**π(SID)(σ(RATING >= 8 ∨ BID = 103)(SAILORS ⨝ RESERVES))

**3. Find the names of sailors who have not reserved a red boat.**π(NAME)(σ(¬(COLOR = 'red'))(SAILORS ⨝ RESERVES ⨝ BOATS))

**4. Find the sailor id’s of sailors with age over 20 who have not reserved a red boat.**π(SID)(σ(AGE > 20 ∧ ¬(COLOR = 'red'))(SAILORS ⨝ RESERVES ⨝ BOATS))

**5. Find the names of sailors who have reserved at least two boats.**π(SNAME)(σ(COUNT(\*) >= 2)(SAILORS ⨝ RESERVES))

**OR**π(SNAME)(ρ(GROUP BY SID HAVING COUNT(\*) >= 2 (RESERVES)) ⨝ SAILORS)

**6. Find the names of sailors who have reserved all boats.**π(sname)(

Sailors ⨝ ρ(sids\_reserved, π(sid)(Reserves)) ÷ ρ(all\_boats, π(bid)(Boats))

(ρ(sailors\_with\_all\_reservations, π(sid, bid)(Reserves)) ))  
  
***Explanation*ρ(sids\_reserved, π(sid)(Reserves)):** This part selects the distinct sailor IDs (sid) from the "Reserves" relation and renames it as "sids\_reserved." It represents the set of sailor IDs that have made reservations.

**ρ(all\_boats, π(bid)(Boats)):** This part selects all distinct boat IDs (bid) from the "Boats" relation and renames it as "all\_boats." It represents the set of all boat IDs.

**ρ(sailors\_with\_all\_reservations, π(sid, bid)(Reserves)):** This part selects the pairs of sailor IDs (sid) and boat IDs (bid) from the "Reserves" relation and renames it as "sailors\_with\_all\_reservations." It represents the set of reservations made by sailors.

**Sailors ⨝ ... (sailors\_with\_all\_reservations):** This performs a natural join between the "Sailors" relation and "sailors\_with\_all\_reservations" to combine information about sailors and their reservations.

**... ÷ ...:** This is the division operator. It divides the result of the join by the Cartesian product of "sids\_reserved" and "all\_boats." This division operation retrieves sailors who have reserved all boats.

**π(sname)(...):** Finally, projection is used to select only the "sname" attribute from the result. This gives us the names of sailors who have reserved all boats.

**7. Find the names of sailors who have reserved all boats called BigBoat.**π(sname)(

Sailors ⨝ ρ(sailors\_with\_all\_bigboat\_reservations, π(sid, bid)(Reserves) ⨝ σ(bname = 'BigBoat')(Boats))

÷ ρ(all\_bigboats, π(bid)(σ(bname = 'BigBoat')(Boats)))

(ρ(all\_bigboats, π(bid)(σ(bname = 'BigBoat')(Boats))))

**8. Find the sailor id’s of sailors whose rating is better than some sailor called Bob.**π(SID)(

σ(rating > (π(rating)(σ(sname = 'Bob')(Sailors))))

(Sailors))

**9. Find the sailor id’s of sailors whose rating is better than every sailor called Bob.**π(SID)(

σ(rating > ALL (π(rating)(σ(sname = 'Bob')(Sailors))))

(Sailors))

**10. Find the sailor id’s of sailors with the highest rating.**

π(SID)(

σ(rating = (π(MAX(rating)(Sailors))))

(Sailors))

**Question 2  
  
1. π(pizzeria) (σ(age < 18) (PERSON ⨝ FREQUENTS))**

**Textual Meaning:**This operation selects the names of pizzerias frequented by persons under the age of 18.

**SQL Query:**SELECT DISTINCT F.pizzeria

FROM Person P

JOIN Frequents F ON P.name = F.name

WHERE P.age < 18;

**2. π(name)(σ(gender = 'female' ∧ (pizza = 'mushroom' ∨ pizza = 'pepperoni')) (PERSON ⨝ EATS))**

**Textual Meaning:** This operation selects the names of persons who are females and have eaten either 'mushroom' or 'pepperoni' pizzas.

**SQL Query:**SELECT DISTINCT P.name

FROM Person P

JOIN Eats E ON P.name = E.name

WHERE P.gender = 'female'

AND (E.pizza = 'mushroom' OR E.pizza = 'pepperoni');  
  
**3. π(pizzeria) (σ(name = 'Amy') (EATS) ⨝ σ(price < 10) (SERVES))**

**Textual Meaning:**This operation selects the names of pizzerias where 'Amy' has eaten pizzas that are served at a price less than 10.

**SQL Query:**SELECT DISTINCT S.pizzeria

FROM Eats E

JOIN Serves S ON E.pizza = S.pizza

WHERE E.name = 'Amy' AND S.price < 10;  
  
**4. π(name) (PERSON) - π(name) (FREQUENTS - π(name, pizzeria) (EATS ⨝ SERVES))**

**Textual Meaning:**This operation selects the names of persons who are not found in the result of (FREQUENTS - (EATS ⨝ SERVES)).

**SQL Query:**SELECT DISTINCT P.name

FROM Person P

WHERE P.name NOT IN (

SELECT F.name

FROM Frequents F

WHERE F.name NOT IN (

SELECT DISTINCT E.name

FROM Eats E

JOIN Serves S ON E.pizza = S.pizza

)

);  
  
**5. π(name) (PERSON) - π(name) (π(name, pizzeria) (EATS ⨝ SERVES) - FREQUENTS)**

**Textual Meaning:** This operation selects the names of persons who are not found in the result of (π(name, pizzeria) (EATS ⨝ SERVES) - FREQUENTS).

**SQL Query:**SELECT DISTINCT P.name

FROM Person P

WHERE P.name NOT IN (

SELECT DISTINCT E.name

FROM Eats E

JOIN Serves S ON E.pizza = S.pizza

EXCEPT

SELECT DISTINCT F.name

FROM Frequents F

);