SQL : Flights (flno, from, to,distance,departs,arrives,price) Aircraft(aid,aname,cruisinrange) Certified (eid,aid) Employees(eid,ename,salary) 1.Find the names of aircraft such that all pilots certified to operate them have salaries more than $80,000. SELECT DISTINCT A.aname FROM Aircraft A WHERE A.Aid IN(SELECTC.aid FROM Certified C, Employees E WHERE C.eid = E.eid AND NOT EXISTS ( SELECT \* FROM Employees E1 WHERE E1.eid = E.eid AND E1.salary < 80000 )) 3. Find the names of pilots whose salary is less than the price of the cheapest route from Los Angeles to Honolulu.

L’algèbre et le calcul relationnel : --the main difference between tuple and domain relationnal calc. Rep : Tuple rel calc = variables range over (is bound to) tuples. Domain relational calculus = variables range over domain variables.-- What a safe relational calculus expression ensures. Rep : Safe relational calculus ensures that a relational calculus expression has a finite set of answers, i.e. a finite set of tuples are returned and all values returned are from the domain of the expression. Safe relational calculus is said to be relationally complete and is equivalent to relational algebra. Contre exemple : (S | not(S € suppliers)) leading to an infinite answer set. For a given set of instance I, a rel. calc. expression Q produces values wich are not in Dom(I,Q) -- Write the relational algebra for : find the names and addresses of the guests who rented the most expensive hotel room available in Ottawa. Rep : ρ(R1, πhotelnum(σcity=`Ottawa`Hotel)) ρ(R2, R1 joint Room) ρ(R3,R2) ρ(R4 (1-> roomno, 2 -> hotelno, 3 -> price), σr3.price < r2.price(R3xR2)) ρ(R5, πhotelno(r2 - πhotelnum,roomno,priceR4) et finalement πguestname,address(Guest joint Booking joint R5) – another ex. Find the sids of suppliers who supply every red part.Rep : RA ( π sid,pid Catalog ) / ( π pid σ color = red Parts ) TRC { T |∃ T 1 ∈ Catalog ( ∀ X ∈ Parts ( X.color ≠red ∨∃ T 2 ∈ Catalog ( T 2 .pid = X.pid ∧ T 2 .sid = T 1 .sid )) ∧ T.sid = T 1 .sid ) } DRC {< X>|< X, Y, Z >∈ Catalog ∧∀< A, B, C>∈ Parts ( C ≠ red ∨∃< P, Q, R >∈ Catalog ( Q = A ∧ P = X )) } SQL SELECT C.sid FROM Catalog C WHERE NOT EXISTS (SELECT P.pid FROM Parts as P WHERE P.color = `red`AND (NOT EXISTS (SELECT C1.sid Catalog C1 WHERE C1.sid = C.sid AND C1.pid = P.pid))) – another ex. Find Find the sids of suppliers who supply some red part and some green part.TRC { T |∃ T 1 ∈ Catalog ( ∃ X ∈ Parts ( X.color = red ∧ X.pid = T 1 .pid ) ∧∃ T 2 ∈ Catalog ( ∃ Y ∈ Parts ( Y.color = green ∧ Y.pid = T 2 .pid ) ∧ T 2 .sid = T 1 .sid ) ∧ T.sid = T 1 .sid ) } DRC {< X>|< X, Y, Z >∈ Catalog ∧∃ A, B, C ( <A, B, C >∈ Parts ∧ C = red ∧ A = Y ) ∧∃ P, Q, R (< P, Q, R >∈ Catalog ∧∃ E,F,G ( <E,F,G>∈ Parts ∧ G = green ∧ E = Q ) ∧ P = X ) }

Formes normales :