

Instruction Session 1: Relational Query Languages Refresher

1. Consider a flight database containing the following information:

- `flight(flight_nr,dept,dest)` is a relation containing the following information about flights: the number of the flight, the departure city and the destination city (e.g. (AA89,BRU,LAX))
- `plane(flight_nr,plane_nr)` is a relation containing the identifiers of the planes that perform certain flights (e.g., (AA89,P1078)). There could be multiple planes for the same flight and vice versa, the same plane could be involved in multiple flights.
- `type(plane_nr,type)` contains for every plane its type (e.g., (P1078,747)).

Formulate the following queries in one of the following three relational query languages:

- algebra (use projection π , selection σ , Cartesian product \times , intersection \cap , union \cup , difference $-$, join \bowtie , renaming ρ)
 - SQL
 - Datalog
- (a) Give all types of planes that are used on a connection from BRU.
 - (b) Give all planes that are not involved in any flight.
 - (c) Give all types of planes that are involved in all flights.
 - (d) Give all connections (i.e. departure-destination pairs) for which not all plane types are used.
 - (e) Give all flight numbers that are used multiple times (e.g., (AA89,BRU,LAX) and (AA89,BRU,JFK) results in (AA89)).
 - (f) Give all pairs of different flight numbers that are used on the same connection (e.g., (AA89,BRU,LAX) and (UA04,BRU,LAX) results in (AA89,UA04)).
 - (g) Give all planes that have a plane type that is used at least once for every connection.

2. True or false? Explain or give a counterexample:

- (a) Every relational algebra expression that does not use negation is monotone; i.e.: if tuples are added to the input, the output can never become smaller.
- (b) The operators \cap and \cup are redundant; i.e., every relational expression can be rewritten into an equivalent one without \cap and \cup .

3. Let G be a relation over the schema (A,B) representing a graph. Express the following query in either SQL or the relational algebra:

- (a) Give all nodes that do not have any incoming edges
- (b) Give all pairs of nodes (x,y) such that together they point to all other nodes; i.e., for every node n in the graph, either (x,n) or (y,n) is in G .
- (c) Give all nodes (x,y) such that the distance (= length of the shortest directed path between them) from x to y is at most 4.