CS186 Discussion Week 5
SQL

- SELECT [DISTINCT] select-list FROM from-list WHERE where-quals GROUP BY grouping-list HAVING group-quals ORDER BY order-list LIMIT max-rows;
- returns multiset, so duplicates rows are possible
- conceptual execution (but inefficient!):
    - compute cross product of all tables in the from-list
    - remove rows that fail where-quals
    - group together all rows which identical values for the columns in grouping-list
    - remove groups which fail the group-quals
    - remove columns which are not in select-list
    - sort the results by the order-list
    - if DISTINCT, remove duplicate rows
    - only return at most max-rows
- union [all], intersect [all], except [all]: [all] keeps around duplicates
- nested subqueries
    - can write another select query in the WHERE clause and use the results as part of a qualification
    - [NOT] IN, [NOT] EXISTS, [NOT] UNIQUE, op ANY, op ALL
    - SELECT * FROM Students WHERE sid NOT IN (SELECT S.sid FROM Students S, Grades G WHERE S.sid = G.sid AND G.grade = “A”);
        - finds students who received no A’s
- Aggregation: aggregates all the values into a single value.
    - COUNT([DISTINCT] col-name), SUM([DISTINCT] col-name), AVG([DISTINCT] col-name), MAX(col-name), MIN(col-name)
- GROUP BY grouping-list
    - grouping-list is list of columns. if select-list has aggs, then any non-agg columns must be in grouping-list
    - groups together all the rows which have the same values of the columns in grouping-list
    - there will be at most 1 result row per group
        - any aggregation in the select-list will be computed over each group, to produce a single value for
    - HAVING group-quals: qualifications for each group, sort of like a WHERE condition for groups each group.
- ORDER BY order-list: sorts the full result in the order of the columns in order-list
- LEFT JOIN: A LEFT JOIN B ON join-conditions
    - if there is no joining rows in B, a row will still be in the result set, but with all the fields of B set to NULL.

Questions

Flights(<u>flight_num</u>, source_city, dest_city)
Departures(<u>flight_num</u>, <u>date</u>, plane_type)
Passengers(<u>passenger_id</u>, passenger_name, passenger_address)
Bookings(<u>passenger_id</u>, <u>flight_num</u>, <u>date</u>, seat_number)

Write SQL expressions for the following queries.

a) Find the passenger id of all passengers who have a seat booked on a plane of type “747” from San Francisco to New York. Do not return any duplicate values.

b) You want to figure out which cities are easy flights to warm weather. Find the cities that have direct (non-stop) flights to both Honolulu and Miami.

c) You want to be the first to book and pick seats for a flight, so find the flight num and date of all flights for which there are no reservations. You also want the results sorted by earliest flight first.

d) Find the number of departures for each type of plane for all flights that leave from Oakland.

e) You want to find the passengers who depart from San Francisco most frequently. Find the top 5 passenger names and the number of their flights which depart from San Francisco. Only include passengers who have departed from San Francisco at least 5 times.