**Query 1 Name:**

`add_airplane()`

**Description:**

This stored procedure creates a new airplane. A new airplane must be sponsored by an existing airline and must have a unique tail number for that airline. An airplane must also have a non-zero seat capacity and speed. An airplane might also have other factors depending on its type, like skids or some number of engines. Finally, new airplanes may or may not have a database-wide unique location identifier but will be given an identifier before people can board it to take a flight.

**Test Case:**

```
call add_airplane('Delta', 'n120jn', 10, 350, NULL, 'jet', NULL, NULL, 4);
```

**Expected Output:**

`Airplane Table:`

| airlineID | tail_num | seat_capacity | speed | locationID | plane_type | skids | propellers | jet_engines |
|---|---|---|---|---|---|---|---|---|
| American | n330ss | 4 | 200 | plane_4 | jet | NULL | NULL | 2 |
| American | n380sd | 5 | 400 | NULL | jet | NULL | NULL | 2 |
| Delta | n106js | 4 | 200 | plane_1 | jet | NULL | NULL | 2 |
| Delta | n110jn | 5 | 600 | plane_2 | jet | NULL | NULL | 4 |
| Delta | n120jn | 10 | 350 | NULL | jet | NULL | NULL | 4 |
| Delta | n127js | 4 | 800 | NULL | NULL | NULL | NULL | NULL |
| Delta | n156sq | 8 | 100 | NULL | NULL | NULL | NULL | NULL |
| JetBlue | n161fk | 4 | 200 | NULL | jet | NULL | NULL | 2 |
| JetBlue | n337as | 5 | 400 | NULL | jet | NULL | NULL | 2 |
| Southwest | n118fm | 4 | 100 | plane_11 | prop | 1 | 1 | NULL |
| Southwest | n401fj | 4 | 200 | plane_9 | jet | NULL | NULL | 2 |
| Southwest | n653fk | 6 | 400 | NULL | jet | NULL | NULL | 2 |
| Southwest | n815pw | 3 | 200 | NULL | prop | 0 | 2 | NULL |
| Spirit | n256ap | 4 | 400 | plane_15 | jet | NULL | NULL | 2 |
| United | n451fi | 5 | 400 | NULL | jet | NULL | NULL | 4 |
| United | n517ly | 4 | 400 | plane_7 | jet | NULL | NULL | 2 |
| United | n616lt | 7 | 400 | NULL | jet | NULL | NULL | 4 |
| United | n620la | 4 | 200 | plane_8 | prop | 0 | 2 | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

**Query 2 Name:**

`add_airport()`

**Description:**

This stored procedure creates a new airport. A new airport must have a unique identifier. New airports may or may not have a database-wide unique location identifier but will be given an identifier before people can go there to catch flights. An airport may have a longer, more descriptive name. An airport must also have a city and state designation.

**Test Case:**

```
call add_airport('SJC', 'San Jose Mineta International Airport',
'San Jose', 'CA', null);
```

**Expected Output:**

`Airport Table:`

| airportID | airport_name | city | state | locationID |
|---|---|---|---|---|
| ABQ | Albuquerque International Sunport | Albuquerque | NM | NULL |
| ANC | Ted Stevens Anchorage International Airport | Anchorage | AK | NULL |
| ATL | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | port_1 |
| BDL | Bradley International Airport | Hartford | CT | NULL |
| BFI | King County International Airport | Seattle | WA | port_10 |
| BHM | Birmingham-Shuttlesworth International Airport | Birmingham | AL | NULL |
| BNA | Nashville International Airport | Nashville | TN | NULL |
| BOI | Boise Airport | Boise | ID | NULL |
| BOS | General Edward Lawrence Logan International Airport | Boston | MA | NULL |
| BTV | Burlington International Airport | Burlington | VT | NULL |
| BWI | Baltimore_Washington International Airport | Baltimore | MD | NULL |
| BZN | Bozeman Yellowstone International Airport | Bozeman | MT | NULL |
| CHS | Charleston International Airport | Charleston | SC | NULL |
| CLE | Cleveland Hopkins International Airport | Cleveland | OH | NULL |
| CLT | Charlotte Douglas International Airport | Charlotte | NC | NULL |
| CRW | Yeager Airport | Charleston | WV | NULL |
| DAL | Dallas Love Field | Dallas | TX | port_7 |
| DCA | Ronald Reagan Washington National Airport | Washington | DC | port_9 |
| DEN | Denver International Airport | Denver | CO | port_3 |
| DFW | Dallas-Fort Worth International Airport | Dallas | TX | port_2 |
| DSM | Des Moines International Airport | Des Moines | IA | NULL |
| DTW | Detroit Metro Wayne County Airport | Detroit | MI | NULL |
| EWR | Newark Liberty International Airport | Newark | NJ | NULL |
| FAR | Hector International Airport | Fargo | ND | NULL |
| FSD | Joe Foss Field | Sioux Falls | SD | NULL |
| GSN | Saipan International Airport | Obyan Saipan Island | MP | NULL |
| GUM | Antonio B_Won Pat International Airport | Agana Tamuning | GU | NULL |
| HNL | Daniel K. Inouye International Airport | Honolulu Oahu | HI | NULL |

| | | | | |
|---|---|---|---|---|
| HOU | William P_Hobby Airport | Houston | TX | port_18 |
| IAD | Washington Dulles International Airport | Washington | DC | port_11 |
| IAH | George Bush Intercontinental Houston Airport | Houston | TX | port_13 |
| ICT | Wichita Dwight D_Eisenhower National Airport | Wichita | KS | NULL |
| ILG | Wilmington Airport | Wilmington | DE | NULL |
| IND | Indianapolis International Airport | Indianapolis | IN | NULL |
| ISP | Long Island MacArthur Airport | New York Islip | NY | port_14 |
| JAC | Jackson Hole Airport | Jackson | WY | NULL |
| JAN | Jackson_Medgar Wiley Evers International Airport | Jackson | MS | NULL |
| JFK | John F_Kennedy International Airport | New York | NY | port_15 |
| LAS | Harry Reid International Airport | Las Vegas | NV | NULL |
| LAX | Los Angeles International Airport | Los Angeles | CA | port_5 |
| LGA | LaGuardia Airport | New York | NY | NULL |
| LIT | Bill and Hillary Clinton National Airport | Little Rock | AR | NULL |
| MCO | Orlando International Airport | Orlando | FL | NULL |
| MDW | Chicago Midway International Airport | Chicago | IL | NULL |
| MHT | Manchester_Boston Regional Airport | Manchester | NH | NULL |
| MKE | Milwaukee Mitchell International Airport | Milwaukee | WI | NULL |
| MRI | Merrill Field | Anchorage | AK | NULL |
| MSP | Minneapolis_St_Paul International Wold_Chamberlain Airport | Minneapolis Saint Paul | MN | NULL |
| MSY | Louis Armstrong New Orleans International Airport | New Orleans | LA | NULL |
| OKC | Will Rogers World Airport | Oklahoma City | OK | NULL |
| OMA | Eppley Airfield | Omaha | NE | NULL |
| ORD | O_Hare International Airport | Chicago | IL | port_4 |
| PDX | Portland International Airport | Portland | OR | NULL |
| PHL | Philadelphia International Airport | Philadelphia | PA | NULL |
| PHX | Phoenix Sky Harbor International Airport | Phoenix | AZ | NULL |
| PVD | Rhode Island T_F_Green International Airport | Providence | RI | NULL |
| PWM | Portland International Jetport | Portland | ME | NULL |
| SDF | Louisville International Airport | Louisville | KY | NULL |
| SEA | Seattle-Tacoma International Airport | Seattle Tacoma | WA | port_17 |
| SJC | San Jose Mineta International Airport | San Jose | CA | NULL |
| SJU | Luis Munoz Marin International Airport | San Juan Carolina | PR | NULL |
| SLC | Salt Lake City International Airport | Salt Lake City | UT | NULL |
| STL | St_Louis Lambert International Airport | Saint Louis | MO | NULL |
| STT | Cyril E_King Airport | Charlotte Amalie Saint Thomas | VI | NULL |
| NULL | NULL | NULL | NULL | NULL |

**Query 3 Name:**

`add_person()`

**Description:**

This stored procedure creates a new person. A new person must have a unique identifier along with a database-wide unique location used to determine where the person is currently located: either at an airport, or on an airplane, at any given time. A person may have a first and last name as well. Also, a person can hold a pilot role, a passenger role, or both roles. As a pilot, a person must have a tax identifier to receive pay, and an experience level. Also, a pilot might be assigned to a specific airplane as part of the flight crew. As a passenger, a person will have some amount of frequent flyer miles.

**Test Case:**

```
call add_person('p78','Sam','Jones','port_2', NULL, 4,
'American', 'n330ss', 20);
```

**Expected Output:**

`Person Table:`

| personID | first_name | last_name | locationID |
|---|---|---|---|
| p1 | Jeanne | Nelson | plane_1 |
| p10 | Lawrence | Morgan | plane_9 |
| p11 | Sandra | Cruz | plane_9 |
| p12 | Dan | Ball | plane_11 |
| p13 | Bryant | Figueroa | plane_2 |
| p14 | Dana | Perry | plane_2 |
| p15 | Matt | Hunt | plane_2 |
| p16 | Edna | Brown | plane_15 |
| p17 | Ruby | Burgess | plane_15 |
| p18 | Esther | Pittman | port_2 |
| p19 | Doug | Fowler | port_4 |
| p2 | Roxanne | Byrd | plane_1 |
| p20 | Thomas | Olson | port_3 |
| p21 | Mona | Harrison | port_4 |
| p22 | Arlene | Massey | port_2 |
| p23 | Judith | Patrick | port_3 |
| p24 | Reginald | Rhodes | plane_1 |
| p25 | Vincent | Garcia | plane_1 |
| p26 | Cheryl | Moore | plane_4 |
| p27 | Michael | Rivera | plane_7 |
| p28 | Luther | Matthews | plane_8 |
| p29 | Moses | Parks | plane_8 |
| p3 | Tanya | Nguyen | plane_4 |
| p30 | Ora | Steele | plane_9 |
| p31 | Antonio | Flores | plane_9 |
| p32 | Glenn | Ross | plane_11 |
| p33 | Irma | Thomas | plane_11 |
| p34 | Ann | Maldonado | plane_2 |
| p35 | Jeffrey | Cruz | plane_2 |
| p36 | Sonya | Price | plane_15 |
| p37 | Tracy | Hale | plane_15 |

| | | | |
|---|---|---|---|
| p38 | Albert | Simmons | port_1 |
| p39 | Karen | Terry | port_9 |
| p4 | Kendra | Jacobs | plane_4 |
| p40 | Glen | Kelley | plane_4 |
| p41 | Brooke | Little | port_4 |
| p42 | Daryl | Nguyen | port_3 |
| p43 | Judy | Willis | port_1 |
| p44 | Marco | Klein | port_2 |
| p45 | Angelica | Hampton | port_5 |
| p5 | Jeff | Burton | plane_4 |
| p6 | Randal | Parks | plane_7 |
| p7 | Sonya | Owens | plane_7 |
| p78 | Sam | Jones | port_2 |
| p8 | Bennie | Palmer | plane_8 |
| p9 | Marlene | Warner | plane_8 |
| NULL | NULL | NULL | NULL |

**Query 4 Name:**

`grant_pilot_license()`

**Description:**

This stored procedure creates a new pilot license. The license must reference a valid pilot and must be a new/unique type of license for that pilot.

**Test Case:**

```
call grant_pilot_license('p1', 'prop');
```

**Expected Output:**

`Pilot_licenses Table`

| personID | license |
|---|---|
| p1 | jet |
| p1 | prop |
| p10 | jet |
| p11 | jet |
| p11 | prop |
| p12 | prop |
| p13 | jet |
| p14 | jet |
| p15 | jet |
| p15 | prop |
| p15 | testing |
| p16 | jet |
| p17 | jet |
| p17 | prop |
| p18 | jet |
| p19 | jet |
| p2 | jet |
| p2 | prop |
| p20 | jet |
| p21 | jet |
| p21 | prop |
| p22 | jet |
| p23 | jet |
| p24 | jet |
| p24 | prop |
| p24 | testing |
| p25 | jet |
| p26 | jet |
| p3 | jet |
| p4 | jet |
| p4 | prop |
| p5 | jet |
| p6 | jet |
| p6 | prop |
| p7 | jet |
| p8 | prop |
| p9 | jet |
| p9 | prop |
| p9 | testing |
| NULL | NULL |

**Query 5 Name:**

`offer_flight()`

**Description:**

This stored procedure creates a new flight. The flight can be defined before an airplane has been assigned for support, but it must have a valid route. Once an airplane has been assigned, we must also track where the airplane is along the route, whether it is in flight or on the ground, and when the next action - takeoff or landing - will occur.

**Test Case:**

```
call offer_flight('UN_3403', 'westbound_north_milk_run',
'American', 'n380sd', 0, 'on_ground', '15:30:00');
```

**Expected Output:**

`Flight Table:`

| flightID | routeID | support_airline | support_tail | progress | airplane_status | next_time |
|---|---|---|---|---|---|---|
| AM_1523 | circle_west_coast | American | n330ss | 2 | on_ground | 14:30:00 |
| DL_1174 | northbound_east_coast | Delta | n106js | 0 | on_ground | 08:00:00 |
| DL_1243 | westbound_north_nonstop | Delta | n110jn | 0 | on_ground | 09:30:00 |
| DL_3410 | circle_east_coast | NULL | NULL | NULL | NULL | NULL |
| SP_1880 | circle_east_coast | Spirit | n256ap | 2 | in_flight | 15:00:00 |
| SW_1776 | hub_xchg_southwest | Southwest | n401fj | 2 | in_flight | 14:00:00 |
| SW_610 | local_texas | Southwest | n118fm | 2 | in_flight | 11:30:00 |
| UN_1899 | eastbound_north_milk_run | United | n517ly | 0 | on_ground | 09:30:00 |
| UN_3403 | westbound_north_milk_run | American | n380sd | 0 | on_ground | 15:30:00 |
| UN_523 | hub_xchg_southeast | United | n620la | 1 | in_flight | 11:00:00 |
| UN_717 | circle_west_coast | NULL | NULL | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

**Query 6 Name:**

`purchase_ticket_and_seat()`

**Description:**

This stored procedure creates a new ticket. The cost of the flight is optional since it might have been a gift, purchased with frequent flyer miles, etc. Each ticket must be tied to a valid person for a valid flight. Also, we will make the (hopefully simplifying) assumption that the departure airport for the ticket will be the airport at which the traveler is currently located. The ticket must also explicitly list the destination airport, which can be an airport before the final airport on the route. Finally, the seat must be unoccupied.

**Test Case:**

```
call purchase_ticket_and_seat('tkt_dl_20', 450, 'DL_1174',
'p23', 'JFK', '5A');
```

**Expected Output:**

`Ticket Table:`

| ticketID | cost | carrier | customer | deplane_at |
|---|---|---|---|---|
| tkt_am_17 | 375 | AM_1523 | p40 | ORD |
| tkt_am_18 | 275 | AM_1523 | p41 | LAX |
| tkt_am_3 | 250 | AM_1523 | p26 | LAX |
| tkt_dl_1 | 450 | DL_1174 | p24 | JFK |
| tkt_dl_11 | 500 | DL_1243 | p34 | LAX |
| tkt_dl_12 | 250 | DL_1243 | p35 | LAX |
| tkt_dl_2 | 225 | DL_1174 | p25 | JFK |
| tkt_dl_20 | 450 | DL_1174 | p23 | JFK |
| tkt_sp_13 | 225 | SP_1880 | p36 | ATL |
| tkt_sp_14 | 150 | SP_1880 | p37 | DCA |
| tkt_sp_16 | 475 | SP_1880 | p39 | ATL |
| tkt_sw_10 | 425 | SW_610 | p33 | HOU |
| tkt_sw_7 | 400 | SW_1776 | p30 | ORD |
| tkt_sw_8 | 175 | SW_1776 | p31 | ORD |
| tkt_sw_9 | 125 | SW_610 | p32 | HOU |
| tkt_un_15 | 150 | UN_523 | p38 | ORD |
| tkt_un_4 | 175 | UN_1899 | p27 | DCA |
| tkt_un_5 | 225 | UN_523 | p28 | ATL |
| tkt_un_6 | 100 | UN_523 | p29 | ORD |
| NULL | NULL | NULL | NULL | NULL |

Ticket_seats Table:

| ticketID | seat_number |
|---|---|
| tkt_am_17 | 2B |
| tkt_am_18 | 2A |
| tkt_am_3 | 3B |
| tkt_dl_1 | 1C |
| tkt_dl_1 | 2F |
| tkt_dl_11 | 1B |
| tkt_dl_11 | 1E |
| tkt_dl_11 | 2F |
| tkt_dl_12 | 2A |
| tkt_dl_2 | 2D |
| tkt_dl_20 | 5A |
| tkt_sp_13 | 1A |
| tkt_sw_10 | 1D |
| tkt_sw_7 | 3C |
| tkt_sw_8 | 3E |
| tkt_sw_9 | 1C |
| tkt_un_4 | 2B |
| tkt_un_5 | 1A |
| tkt_un_6 | 3B |
| NULL | NULL |

**Query 7 Name:**

`add_update_leg()`

**Description:**

This stored procedure creates a new leg as specified. However, if a leg from the departure airport to the arrival airport already exists, then don't create a new leg - instead, update the existence of the current leg while keeping the existing identifier. Also, all legs must be symmetric. If a leg in the opposite direction exists, then update the distance to ensure that it is equivalent.

**Test Case:**

```
call add_update_leg('leg_28', 2800, 'DCA', 'SEA');
```

**Expected Output:**

`Leg Table:`

| legID | distance | departure | arrival |
|---|---|---|---|
| leg_1 | 600 | ATL | IAD |
| leg_10 | 800 | DFW | ORD |
| leg_11 | 600 | IAD | ORD |
| leg_12 | 200 | IAH | DAL |
| leg_13 | 1400 | IAH | LAX |
| leg_14 | 2400 | ISP | BFI |
| leg_15 | 800 | JFK | ATL |
| leg_16 | 800 | JFK | ORD |
| leg_17 | 2400 | JFK | SEA |
| leg_18 | 1200 | LAX | DFW |
| leg_19 | 1000 | LAX | SEA |
| leg_2 | 600 | ATL | IAH |
| leg_20 | 600 | ORD | DCA |
| leg_21 | 800 | ORD | DFW |
| leg_22 | 800 | ORD | LAX |
| leg_23 | 2400 | SEA | JFK |
| leg_24 | 1800 | SEA | ORD |
| leg_25 | 600 | ORD | ATL |
| leg_26 | 800 | LAX | ORD |
| leg_27 | 1600 | ATL | LAX |
| leg_28 | 2800 | DCA | SEA |
| leg_3 | 800 | ATL | JFK |
| leg_4 | 600 | ATL | ORD |
| leg_5 | 1000 | BFI | LAX |
| leg_6 | 200 | DAL | HOU |
| leg_7 | 600 | DCA | ATL |
| leg_8 | 200 | DCA | JFK |
| leg_9 | 800 | DFW | ATL |
| NULL | NULL | NULL | NULL |

**Query 8 Name:**

`start_route()`

**Description:**

This stored procedure creates the first leg of a new route. Routes in our system must be created in the sequential order of the legs. The first leg of the route can be any valid leg, and it should have a sequence number of 1.

**Test Case:**

```
call start_route('new_eastbound_west_milk_run','leg_10');
```

**Expected Output:**

`Route Table:`

| routeID |
| --- |
| circle_east_coast |
| circle_west_coast |
| eastbound_north_milk_run |
| eastbound_north_nonstop |
| eastbound_south_milk_run |
| hub_xchg_southeast |
| hub_xchg_southwest |
| local_texas |
| new_eastbound_west_milk_run |
| northbound_east_coast |
| northbound_west_coast |
| southbound_midwest |
| westbound_north_milk_run |
| westbound_north_nonstop |
| westbound_south_nonstop |
| NULL |

`Route_path Table:`

| routeID | legID | sequence |
|---|---|---|
| eastbound_south_milk_run | leg_1 | 3 |
| circle_west_coast | leg_10 | 2 |
| new_eastbound_west_milk_run | leg_10 | 1 |
| local_texas | leg_12 | 1 |
| westbound_north_milk_run | leg_16 | 1 |
| westbound_north_nonstop | leg_17 | 1 |
| circle_west_coast | leg_18 | 1 |
| eastbound_south_milk_run | leg_18 | 1 |
| northbound_west_coast | leg_19 | 1 |
| westbound_north_milk_run | leg_19 | 3 |
| circle_east_coast | leg_20 | 2 |
| eastbound_north_milk_run | leg_20 | 2 |
| southbound_midwest | leg_21 | 1 |
| circle_west_coast | leg_22 | 3 |
| hub_xchg_southwest | leg_22 | 1 |
| westbound_north_milk_run | leg_22 | 2 |
| eastbound_north_nonstop | leg_23 | 1 |
| eastbound_north_milk_run | leg_24 | 1 |
| hub_xchg_southeast | leg_25 | 1 |
| hub_xchg_southwest | leg_26 | 2 |
| westbound_south_nonstop | leg_27 | 1 |
| northbound_east_coast | leg_3 | 1 |
| circle_east_coast | leg_4 | 1 |
| hub_xchg_southeast | leg_4 | 2 |
| local_texas | leg_6 | 2 |
| circle_east_coast | leg_7 | 3 |
| eastbound_north_milk_run | leg_8 | 3 |
| eastbound_south_milk_run | leg_9 | 2 |
| NULL | NULL | NULL |

**Query 9 Name:**

`extend_route()`

**Description:**

This stored procedure adds another leg to the end of an existing route. Routes in our system must be created in the sequential order of the legs, and the route must be contiguous: the departure airport of this leg must be the same as the arrival airport of the previous leg.

**Test Case:**

```
call extend_route('eastbound_south_milk_run', 'leg_11');
```

**Expected Output:**

`Route Table:`

| routeID |
| --- |
| circle_east_coast |
| circle_west_coast |
| eastbound_north_milk_run |
| eastbound_north_nonstop |
| eastbound_south_milk_run |
| hub_xchg_southeast |
| hub_xchg_southwest |
| local_texas |
| northbound_east_coast |
| northbound_west_coast |
| southbound_midwest |
| westbound_north_milk_run |
| westbound_north_nonstop |
| westbound_south_nonstop |
| NULL |

`Route_path Table:`

| routeID | legID | sequence |
|---|---|---|
| eastbound_south_milk_run | leg_1 | 3 |
| circle_west_coast | leg_10 | 2 |
| eastbound_south_milk_run | leg_11 | 4 |
| local_texas | leg_12 | 1 |
| westbound_north_milk_run | leg_16 | 1 |
| westbound_north_nonstop | leg_17 | 1 |
| circle_west_coast | leg_18 | 1 |
| eastbound_south_milk_run | leg_18 | 1 |
| northbound_west_coast | leg_19 | 1 |
| westbound_north_milk_run | leg_19 | 3 |
| circle_east_coast | leg_20 | 2 |
| eastbound_north_milk_run | leg_20 | 2 |
| southbound_midwest | leg_21 | 1 |
| circle_west_coast | leg_22 | 3 |
| hub_xchg_southwest | leg_22 | 1 |
| westbound_north_milk_run | leg_22 | 2 |
| eastbound_north_nonstop | leg_23 | 1 |
| eastbound_north_milk_run | leg_24 | 1 |
| hub_xchg_southeast | leg_25 | 1 |
| hub_xchg_southwest | leg_26 | 2 |
| westbound_south_nonstop | leg_27 | 1 |
| northbound_east_coast | leg_3 | 1 |
| circle_east_coast | leg_4 | 1 |
| hub_xchg_southeast | leg_4 | 2 |
| local_texas | leg_6 | 2 |
| circle_east_coast | leg_7 | 3 |
| eastbound_north_milk_run | leg_8 | 3 |
| eastbound_south_milk_run | leg_9 | 2 |
| NULL | NULL | NULL |

**Query 10 Name:**

`flight_landing()`

**Description:**

This stored procedure updates the state for a flight landing at the next airport along its route. The time for the flight should be moved one hour into the future to allow for the flight to be checked, refueled, restocked, etc. for the next leg of travel. Also, the pilots of the flight should receive increased experience, and the passengers should have their frequent flyer miles updated.

**Test Case:**

```
call flight_landing('SW_1776');
```

**Expected Output:**

`Flight Table:`

| flightID | routeID | support_airline | support_tail | progress | airplane_status | next_time |
|---|---|---|---|---|---|---|
| AM_1523 | circle_west_coast | American | n330ss | 2 | on_ground | 14:30:00 |
| DL_1174 | northbound_east_coast | Delta | n106js | 0 | on_ground | 08:00:00 |
| DL_1243 | westbound_north_nonstop | Delta | n110jn | 0 | on_ground | 09:30:00 |
| DL_3410 | circle_east_coast | NULL | NULL | NULL | NULL | NULL |
| SP_1880 | circle_east_coast | Spirit | n256ap | 2 | in_flight | 15:00:00 |
| SW_1776 | hub_xchg_southwest | Southwest | n401fj | 2 | on_ground | 15:00:00 |
| SW_610 | local_texas | Southwest | n118fm | 2 | in_flight | 11:30:00 |
| UN_1899 | eastbound_north_milk_run | United | n517ly | 0 | on_ground | 09:30:00 |
| UN_523 | hub_xchg_southeast | United | n620la | 1 | in_flight | 11:00:00 |
| UN_717 | circle_west_coast | NULL | NULL | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

`Pilot Table:`

| personID | taxID | experience | flying_airline | flying_tail |
|---|---|---|---|---|
| p1 | 330-12-6907 | 31 | Delta | n106js |
| p10 | 769-60-1266 | 16 | Southwest | n401fj |
| p11 | 369-22-9505 | 23 | Southwest | n401fj |
| p12 | 680-92-5329 | 24 | Southwest | n118fm |
| p13 | 513-40-4168 | 24 | Delta | n110jn |
| p14 | 454-71-7847 | 13 | Delta | n110jn |
| p15 | 153-47-8101 | 30 | Delta | n110jn |
| p16 | 598-47-5172 | 28 | Spirit | n256ap |
| p17 | 865-71-6800 | 36 | Spirit | n256ap |
| p18 | 250-86-2784 | 23 | NULL | NULL |
| p19 | 386-39-7881 | 2 | NULL | NULL |
| p2 | 842-88-1257 | 9 | Delta | n106js |
| p20 | 522-44-3098 | 28 | NULL | NULL |
| p21 | 621-34-5755 | 2 | NULL | NULL |
| p22 | 177-47-9877 | 3 | NULL | NULL |
| p23 | 528-64-7912 | 12 | NULL | NULL |
| p24 | 803-30-1789 | 34 | NULL | NULL |
| p25 | 986-76-1587 | 13 | NULL | NULL |
| p26 | 584-77-5105 | 20 | NULL | NULL |
| p3 | 750-24-7616 | 11 | American | n330ss |
| p4 | 776-21-8098 | 24 | American | n330ss |
| p5 | 933-93-2165 | 27 | American | n330ss |
| p6 | 707-84-4555 | 38 | United | n517ly |
| p7 | 450-25-5617 | 13 | United | n517ly |
| p8 | 701-38-2179 | 12 | United | n620la |
| p9 | 936-44-6941 | 13 | United | n620la |
| NULL | NULL | NULL | NULL | NULL |

Passenger Table:

| personID | miles |
|---|---|
| p21 | 771 |
| p22 | 374 |
| p23 | 414 |
| p24 | 292 |
| p25 | 390 |
| p26 | 302 |
| p27 | 470 |
| p28 | 208 |
| p29 | 292 |
| p30 | 1486 |
| p31 | 1347 |
| p32 | 257 |
| p33 | 564 |
| p34 | 211 |
| p35 | 233 |
| p36 | 293 |
| p37 | 552 |
| p38 | 812 |
| p39 | 541 |
| p40 | 441 |
| p41 | 875 |
| p42 | 691 |
| p43 | 572 |
| p44 | 572 |
| p45 | 663 |
| NULL | NULL |

**Query 11 Name:**

`flight_takeoff()`

**Description:**

This stored procedure updates the state for a flight taking off from its current airport towards the next airport along its route. The time for the next time of the flight must be calculated based on the distance and the speed of the airplane. And we must also ensure that propeller driven planes have at least one pilot assigned, while jets must have a minimum of two pilots. If the flight cannot take off because of a pilot shortage, then the flight must be delayed for 30 minutes.

**Test Case:**

```
call flight_takeoff('DL_1174');
```

**Expected Output:**

`Flight Table:`

| flightID | routeID | support_airline | support_tail | progress | airplane_status | next_time |
|---|---|---|---|---|---|---|
| AM_1523 | circle_west_coast | American | n330ss | 2 | on_ground | 14:30:00 |
| DL_1174 | northbound_east_coast | Delta | n106js | 1 | in_flight | 12:00:00 |
| DL_1243 | westbound_north_nonstop | Delta | n110jn | 0 | on_ground | 09:30:00 |
| DL_3410 | circle_east_coast | NULL | NULL | NULL | NULL | NULL |
| SP_1880 | circle_east_coast | Spirit | n256ap | 2 | in_flight | 15:00:00 |
| SW_1776 | hub_xchg_southwest | Southwest | n401fj | 2 | in_flight | 14:00:00 |
| SW_610 | local_texas | Southwest | n118fm | 2 | in_flight | 11:30:00 |
| UN_1899 | eastbound_north_milk_run | United | n517ly | 0 | on_ground | 09:30:00 |
| UN_523 | hub_xchg_southeast | United | n620la | 1 | in_flight | 11:00:00 |
| UN_717 | circle_west_coast | NULL | NULL | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

**Query 12 Name:**

passengers_board()

**Description:**

This stored procedure updates the state for passengers getting on a flight at its current airport. The passengers must be at the airport and hold a valid ticket for the flight.

**Test Case:**

```
call passengers_board('AM_1523');
```

**Expected Output:**

Person Table:

| personID | first_name | last_name | locationID |
|---|---|---|---|
| p1 | Jeanne | Nelson | plane_1 |
| p10 | Lawrence | Morgan | plane_9 |
| p11 | Sandra | Cruz | plane_9 |
| p12 | Dan | Ball | plane_11 |
| p13 | Bryant | Figueroa | plane_2 |
| p14 | Dana | Perry | plane_2 |
| p15 | Matt | Hunt | plane_2 |
| p16 | Edna | Brown | plane_15 |
| p17 | Ruby | Burgess | plane_15 |
| p18 | Esther | Pittman | port_2 |
| p19 | Doug | Fowler | port_4 |
| p2 | Roxanne | Byrd | plane_1 |
| p20 | Thomas | Olson | port_3 |
| p21 | Mona | Harrison | port_4 |
| p22 | Arlene | Massey | port_2 |
| p23 | Judith | Patrick | port_3 |
| p24 | Reginald | Rhodes | plane_1 |
| p25 | Vincent | Garcia | plane_1 |
| p26 | Cheryl | Moore | plane_4 |
| p27 | Michael | Rivera | plane_7 |
| p28 | Luther | Matthews | plane_8 |
| p29 | Moses | Parks | plane_8 |
| p3 | Tanya | Nguyen | plane_4 |
| p30 | Ora | Steele | plane_9 |
| p31 | Antonio | Flores | plane_9 |
| p32 | Glenn | Ross | plane_11 |
| p33 | Irma | Thomas | plane_11 |
| p34 | Ann | Maldonado | plane_2 |
| p35 | Jeffrey | Cruz | plane_2 |
| p36 | Sonya | Price | plane_15 |
| p37 | Tracy | Hale | plane_15 |
| p38 | Albert | Simmons | port_1 |
| p39 | Karen | Terry | port_9 |
| p4 | Kendra | Jacobs | plane_4 |
| p40 | Glen | Kelley | plane_4 |
| p41 | Brooke | Little | plane_4 |
| p42 | Daryl | Nguyen | port_3 |
| p43 | Judy | Willis | port_1 |
| p44 | Marco | Klein | port_2 |
| p45 | Angelica | Hampton | port_5 |
| p5 | Jeff | Burton | plane_4 |
| p6 | Randal | Parks | plane_7 |
| p7 | Sonya | Owens | plane_7 |
| p8 | Bennie | Palmer | plane_8 |
| p9 | Marlene | Warner | plane_8 |
| NULL | NULL | NULL | NULL |

**Query 13 Name:**

`passengers_disembark()`

**Description:**

This stored procedure updates the state for passengers getting off a flight at its current airport. The passengers must be on that flight, and the flight must be located at the destination airport as referenced by the ticket.

**Test Case:**

```
call passengers_disembark('AM_1523');
```

**Expected Output:**

`Person Table:`

| personID | first_name | last_name | locationID |
|---|---|---|---|
| p1 | Jeanne | Nelson | plane_1 |
| p10 | Lawrence | Morgan | plane_9 |
| p11 | Sandra | Cruz | plane_9 |
| p12 | Dan | Ball | plane_11 |
| p13 | Bryant | Figueroa | plane_2 |
| p14 | Dana | Perry | plane_2 |
| p15 | Matt | Hunt | plane_2 |
| p16 | Edna | Brown | plane_15 |
| p17 | Ruby | Burgess | plane_15 |
| p18 | Esther | Pittman | port_2 |
| p19 | Doug | Fowler | port_4 |
| p2 | Roxanne | Byrd | plane_1 |
| p20 | Thomas | Olson | port_3 |
| p21 | Mona | Harrison | port_4 |
| p22 | Arlene | Massey | port_2 |
| p23 | Judith | Patrick | port_3 |
| p24 | Reginald | Rhodes | plane_1 |
| p25 | Vincent | Garcia | plane_1 |
| p26 | Cheryl | Moore | plane_4 |
| p27 | Michael | Rivera | plane_7 |
| p28 | Luther | Matthews | plane_8 |
| p29 | Moses | Parks | plane_8 |
| p3 | Tanya | Nguyen | plane_4 |
| p30 | Ora | Steele | plane_9 |
| p31 | Antonio | Flores | plane_9 |
| p32 | Glenn | Ross | plane_11 |
| p33 | Irma | Thomas | plane_11 |
| p34 | Ann | Maldonado | plane_2 |
| p35 | Jeffrey | Cruz | plane_2 |
| p36 | Sonya | Price | plane_15 |
| p37 | Tracy | Hale | plane_15 |
| p38 | Albert | Simmons | port_1 |
| p39 | Karen | Terry | port_9 |
| p4 | Kendra | Jacobs | plane_4 |
| p40 | Glen | Kelley | port_4 |
| p41 | Brooke | Little | port_4 |
| p42 | Daryl | Nguyen | port_3 |
| p43 | Judy | Willis | port_1 |
| p44 | Marco | Klein | port_2 |
| p45 | Angelica | Hampton | port_5 |
| p5 | Jeff | Burton | plane_4 |
| p6 | Randal | Parks | plane_7 |
| p7 | Sonya | Owens | plane_7 |
| p8 | Bennie | Palmer | plane_8 |
| p9 | Marlene | Warner | plane_8 |
| NULL | NULL | NULL | NULL |

**Query 14 Name:**

`assign_pilot()`

**Description:**

This stored procedure assigns a pilot as part of the flight crew for a given airplane. The pilot being assigned must have a license for that type of airplane, and must be at the same location as the flight. Also, a pilot can only support one flight (i.e. one airplane) at a time. The pilot must be assigned to the flight and have their location updated for the appropriate airplane.

**Test Case:**

```
call assign_pilot('AM_1523', 'p19');
```

**Expected Output:**

`Pilot Table:`

| personID | taxID | experience | flying_airline | flying_tail |
|---|---|---|---|---|
| p1 | 330-12-6907 | 31 | Delta | n106js |
| p10 | 769-60-1266 | 15 | Southwest | n401fj |
| p11 | 369-22-9505 | 22 | Southwest | n401fj |
| p12 | 680-92-5329 | 24 | Southwest | n118fm |
| p13 | 513-40-4168 | 24 | Delta | n110jn |
| p14 | 454-71-7847 | 13 | Delta | n110jn |
| p15 | 153-47-8101 | 30 | Delta | n110jn |
| p16 | 598-47-5172 | 28 | Spirit | n256ap |
| p17 | 865-71-6800 | 36 | Spirit | n256ap |
| p18 | 250-86-2784 | 23 | NULL | NULL |
| p19 | 386-39-7881 | 2 | American | n330ss |
| p2 | 842-88-1257 | 9 | Delta | n106js |
| p20 | 522-44-3098 | 28 | NULL | NULL |
| p21 | 621-34-5755 | 2 | NULL | NULL |
| p22 | 177-47-9877 | 3 | NULL | NULL |
| p23 | 528-64-7912 | 12 | NULL | NULL |
| p24 | 803-30-1789 | 34 | NULL | NULL |
| p25 | 986-76-1587 | 13 | NULL | NULL |
| p26 | 584-77-5105 | 20 | NULL | NULL |
| p3 | 750-24-7616 | 11 | American | n330ss |
| p4 | 776-21-8098 | 24 | American | n330ss |
| p5 | 933-93-2165 | 27 | American | n330ss |
| p6 | 707-84-4555 | 38 | United | n517ly |
| p7 | 450-25-5617 | 13 | United | n517ly |
| p8 | 701-38-2179 | 12 | United | n620la |
| p9 | 936-44-6941 | 13 | United | n620la |
| NULL | NULL | NULL | NULL | NULL |

Person Table:

| personID | first_name | last_name | locationID |
|---|---|---|---|
| p1 | Jeanne | Nelson | plane_1 |
| p10 | Lawrence | Morgan | plane_9 |
| p11 | Sandra | Cruz | plane_9 |
| p12 | Dan | Ball | plane_11 |
| p13 | Bryant | Figueroa | plane_2 |
| p14 | Dana | Perry | plane_2 |
| p15 | Matt | Hunt | plane_2 |
| p16 | Edna | Brown | plane_15 |
| p17 | Ruby | Burgess | plane_15 |
| p18 | Esther | Pittman | port_2 |
| p19 | Doug | Fowler | plane_4 |
| p2 | Roxanne | Byrd | plane_1 |
| p20 | Thomas | Olson | port_3 |
| p21 | Mona | Harrison | port_4 |
| p22 | Arlene | Massey | port_2 |
| p23 | Judith | Patrick | port_3 |
| p24 | Reginald | Rhodes | plane_1 |
| p25 | Vincent | Garcia | plane_1 |
| p26 | Cheryl | Moore | plane_4 |
| p27 | Michael | Rivera | plane_7 |
| p28 | Luther | Matthews | plane_8 |
| p29 | Moses | Parks | plane_8 |
| p3 | Tanya | Nguyen | plane_4 |
| p30 | Ora | Steele | plane_9 |
| p31 | Antonio | Flores | plane_9 |
| p32 | Glenn | Ross | plane_11 |
| p33 | Irma | Thomas | plane_11 |
| p34 | Ann | Maldonado | plane_2 |
| p35 | Jeffrey | Cruz | plane_2 |
| p36 | Sonya | Price | plane_15 |
| p37 | Tracy | Hale | plane_15 |
| p38 | Albert | Simmons | port_1 |
| p39 | Karen | Terry | port_9 |
| p4 | Kendra | Jacobs | plane_4 |
| p40 | Glen | Kelley | plane_4 |
| p41 | Brooke | Little | port_4 |
| p42 | Daryl | Nguyen | port_3 |
| p43 | Judy | Willis | port_1 |
| p44 | Marco | Klein | port_2 |
| p45 | Angelica | Hampton | port_5 |
| p5 | Jeff | Burton | plane_4 |
| p6 | Randal | Parks | plane_7 |
| p7 | Sonya | Owens | plane_7 |
| p8 | Bennie | Palmer | plane_8 |
| p9 | Marlene | Warner | plane_8 |
| NULL | NULL | NULL | NULL |

**Query 15 Name:**

`recycle_crew()`

**Description:**

This stored procedure releases the assignments for a given flight crew. The flight must have ended, and all passengers must have disembarked.

**Test Case (You must run all 3 statements in the given order):**

```
update flight set progress = 3 where flightID = 'AM_1523';
update person set locationID = 'port_5' where personID in
('p26', 'p40', 'p41');
call recycle_crew('AM_1523');
```

**Expected Output:**

`Pilot Table:`

| personID | taxID | experience | flying_airline | flying_tail |
|---|---|---|---|---|
| p1 | 330-12-6907 | 31 | Delta | n106js |
| p10 | 769-60-1266 | 15 | Southwest | n401fj |
| p11 | 369-22-9505 | 22 | Southwest | n401fj |
| p12 | 680-92-5329 | 24 | Southwest | n118fm |
| p13 | 513-40-4168 | 24 | Delta | n110jn |
| p14 | 454-71-7847 | 13 | Delta | n110jn |
| p15 | 153-47-8101 | 30 | Delta | n110jn |
| p16 | 598-47-5172 | 28 | Spirit | n256ap |
| p17 | 865-71-6800 | 36 | Spirit | n256ap |
| p18 | 250-86-2784 | 23 | NULL | NULL |
| p19 | 386-39-7881 | 2 | NULL | NULL |
| p2 | 842-88-1257 | 9 | Delta | n106js |
| p20 | 522-44-3098 | 28 | NULL | NULL |
| p21 | 621-34-5755 | 2 | NULL | NULL |
| p22 | 177-47-9877 | 3 | NULL | NULL |
| p23 | 528-64-7912 | 12 | NULL | NULL |
| p24 | 803-30-1789 | 34 | NULL | NULL |
| p25 | 986-76-1587 | 13 | NULL | NULL |
| p26 | 584-77-5105 | 20 | NULL | NULL |
| p3 | 750-24-7616 | 11 | NULL | NULL |
| p4 | 776-21-8098 | 24 | NULL | NULL |
| p5 | 933-93-2165 | 27 | NULL | NULL |
| p6 | 707-84-4555 | 38 | United | n517ly |
| p7 | 450-25-5617 | 13 | United | n517ly |
| p8 | 701-38-2179 | 12 | United | n620la |
| p9 | 936-44-6941 | 13 | United | n620la |
| NULL | NULL | NULL | NULL | NULL |

Person Table:

| personID | first_name | last_name | locationID |
|---|---|---|---|
| p1 | Jeanne | Nelson | plane_1 |
| p10 | Lawrence | Morgan | plane_9 |
| p11 | Sandra | Cruz | plane_9 |
| p12 | Dan | Ball | plane_11 |
| p13 | Bryant | Figueroa | plane_2 |
| p14 | Dana | Perry | plane_2 |
| p15 | Matt | Hunt | plane_2 |
| p16 | Edna | Brown | plane_15 |
| p17 | Ruby | Burgess | plane_15 |
| p18 | Esther | Pittman | port_2 |
| p19 | Doug | Fowler | port_4 |
| p2 | Roxanne | Byrd | plane_1 |
| p20 | Thomas | Olson | port_3 |
| p21 | Mona | Harrison | port_4 |
| p22 | Arlene | Massey | port_2 |
| p23 | Judith | Patrick | port_3 |
| p24 | Reginald | Rhodes | plane_1 |
| p25 | Vincent | Garcia | plane_1 |
| p26 | Cheryl | Moore | port_5 |
| p27 | Michael | Rivera | plane_7 |
| p28 | Luther | Matthews | plane_8 |
| p29 | Moses | Parks | plane_8 |
| p3 | Tanya | Nguyen | port_5 |
| p30 | Ora | Steele | plane_9 |
| p31 | Antonio | Flores | plane_9 |
| p32 | Glenn | Ross | plane_11 |
| p33 | Irma | Thomas | plane_11 |
| p34 | Ann | Maldonado | plane_2 |
| p35 | Jeffrey | Cruz | plane_2 |
| p36 | Sonya | Price | plane_15 |
| p37 | Tracy | Hale | plane_15 |
| p38 | Albert | Simmons | port_1 |
| p39 | Karen | Terry | port_9 |
| p4 | Kendra | Jacobs | port_5 |
| p40 | Glen | Kelley | port_5 |
| p41 | Brooke | Little | port_5 |
| p42 | Daryl | Nguyen | port_3 |
| p43 | Judy | Willis | port_1 |
| p44 | Marco | Klein | port_2 |
| p45 | Angelica | Hampton | port_5 |
| p5 | Jeff | Burton | port_5 |
| p6 | Randal | Parks | plane_7 |
| p7 | Sonya | Owens | plane_7 |
| p8 | Bennie | Palmer | plane_8 |
| p9 | Marlene | Warner | plane_8 |
| NULL | NULL | NULL | NULL |

**Query 16 Name:**

`retire_flight()`

**Description:**

This stored procedure removes a flight that has ended from the system. The flight must be on the ground, and either be at the start of its route or at the end of its route.

**Test Case:**

```
call retire_flight('DL_1243');
```

**Expected Output:**

`Flight Table:`

| flightID | routeID | support_airline | support_tail | progress | airplane_status | next_time |
|---|---|---|---|---|---|---|
| AM_1523 | circle_west_coast | American | n330ss | 2 | on_ground | 14:30:00 |
| DL_1174 | northbound_east_coast | Delta | n106js | 0 | on_ground | 08:00:00 |
| DL_3410 | circle_east_coast | NULL | NULL | NULL | NULL | NULL |
| SP_1880 | circle_east_coast | Spirit | n256ap | 2 | in_flight | 15:00:00 |
| SW_1776 | hub_xchg_southwest | Southwest | n401fj | 2 | in_flight | 14:00:00 |
| SW_610 | local_texas | Southwest | n118fm | 2 | in_flight | 11:30:00 |
| UN_1899 | eastbound_north_milk_run | United | n517ly | 0 | on_ground | 09:30:00 |
| UN_523 | hub_xchg_southeast | United | n620la | 1 | in_flight | 11:00:00 |
| UN_717 | circle_west_coast | NULL | NULL | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

**Query 17 Name:**

`remove_passenger_role()`

**Description:**

This stored procedure removes the passenger role from person. The passenger must be on the ground at the time; and, if they are on a flight, then they must disembark the flight at the current airport. If the person had both a pilot role and a passenger role, then the person and pilot role data should not be affected. If the person only had a passenger role, then all associated person data must be removed as well.

**Test Case:**

```
call remove_passenger_role('p45');
```

**Expected Output:**

`Passenger Table:`

| personID | miles |
|---|---|
| p21 | 771 |
| p22 | 374 |
| p23 | 414 |
| p24 | 292 |
| p25 | 390 |
| p26 | 302 |
| p27 | 470 |
| p28 | 208 |
| p29 | 292 |
| p30 | 686 |
| p31 | 547 |
| p32 | 257 |
| p33 | 564 |
| p34 | 211 |
| p35 | 233 |
| p36 | 293 |
| p37 | 552 |
| p38 | 812 |
| p39 | 541 |
| p40 | 441 |
| p41 | 875 |
| p42 | 691 |
| p43 | 572 |
| p44 | 572 |
| NULL | NULL |

Person Table:

| personID | first_name | last_name | locationID |
|---|---|---|---|
| p1 | Jeanne | Nelson | plane_1 |
| p10 | Lawrence | Morgan | plane_9 |
| p11 | Sandra | Cruz | plane_9 |
| p12 | Dan | Ball | plane_11 |
| p13 | Bryant | Figueroa | plane_2 |
| p14 | Dana | Perry | plane_2 |
| p15 | Matt | Hunt | plane_2 |
| p16 | Edna | Brown | plane_15 |
| p17 | Ruby | Burgess | plane_15 |
| p18 | Esther | Pittman | port_2 |
| p19 | Doug | Fowler | port_4 |
| p2 | Roxanne | Byrd | plane_1 |
| p20 | Thomas | Olson | port_3 |
| p21 | Mona | Harrison | port_4 |
| p22 | Arlene | Massey | port_2 |
| p23 | Judith | Patrick | port_3 |
| p24 | Reginald | Rhodes | plane_1 |
| p25 | Vincent | Garcia | plane_1 |
| p26 | Cheryl | Moore | plane_4 |
| p27 | Michael | Rivera | plane_7 |
| p28 | Luther | Matthews | plane_8 |
| p29 | Moses | Parks | plane_8 |
| p3 | Tanya | Nguyen | plane_4 |
| p30 | Ora | Steele | plane_9 |
| p31 | Antonio | Flores | plane_9 |
| p32 | Glenn | Ross | plane_11 |
| p33 | Irma | Thomas | plane_11 |
| p34 | Ann | Maldonado | plane_2 |
| p35 | Jeffrey | Cruz | plane_2 |
| p36 | Sonya | Price | plane_15 |
| p37 | Tracy | Hale | plane_15 |
| p38 | Albert | Simmons | port_1 |
| p39 | Karen | Terry | port_9 |
| p4 | Kendra | Jacobs | plane_4 |
| p40 | Glen | Kelley | plane_4 |
| p41 | Brooke | Little | port_4 |
| p42 | Daryl | Nguyen | port_3 |
| p43 | Judy | Willis | port_1 |
| p44 | Marco | Klein | port_2 |
| p5 | Jeff | Burton | plane_4 |
| p6 | Randal | Parks | plane_7 |
| p7 | Sonya | Owens | plane_7 |
| p8 | Bennie | Palmer | plane_8 |
| p9 | Marlene | Warner | plane_8 |
| NULL | NULL | NULL | NULL |

**Query 18 Name:**

`remove_pilot_role()`

**Description:**

This stored procedure removes the pilot role from person. The pilot must not be assigned to a flight; or, if they are assigned to a flight, then that flight must either be at the start or end of its route. If the person had both a pilot role and a passenger role, then the person and passenger role data should not be affected. If the person only had a pilot role, then all associated person data must be removed as well.

**Test Case:**

```
call remove_pilot_role('p20');
```

**Expected Output:**

`Pilot Table:`

| personID | taxID | experience | flying_airline | flying_tail |
|---|---|---|---|---|
| p1 | 330-12-6907 | 31 | Delta | n106js |
| p10 | 769-60-1266 | 15 | Southwest | n401fj |
| p11 | 369-22-9505 | 22 | Southwest | n401fj |
| p12 | 680-92-5329 | 24 | Southwest | n118fm |
| p13 | 513-40-4168 | 24 | Delta | n110jn |
| p14 | 454-71-7847 | 13 | Delta | n110jn |
| p15 | 153-47-8101 | 30 | Delta | n110jn |
| p16 | 598-47-5172 | 28 | Spirit | n256ap |
| p17 | 865-71-6800 | 36 | Spirit | n256ap |
| p18 | 250-86-2784 | 23 | NULL | NULL |
| p19 | 386-39-7881 | 2 | NULL | NULL |
| p2 | 842-88-1257 | 9 | Delta | n106js |
| p21 | 621-34-5755 | 2 | NULL | NULL |
| p22 | 177-47-9877 | 3 | NULL | NULL |
| p23 | 528-64-7912 | 12 | NULL | NULL |
| p24 | 803-30-1789 | 34 | NULL | NULL |
| p25 | 986-76-1587 | 13 | NULL | NULL |
| p26 | 584-77-5105 | 20 | NULL | NULL |
| p3 | 750-24-7616 | 11 | American | n330ss |
| p4 | 776-21-8098 | 24 | American | n330ss |
| p5 | 933-93-2165 | 27 | American | n330ss |
| p6 | 707-84-4555 | 38 | United | n517ly |
| p7 | 450-25-5617 | 13 | United | n517ly |
| p8 | 701-38-2179 | 12 | United | n620la |
| p9 | 936-44-6941 | 13 | United | n620la |
| NULL | NULL | NULL | NULL | NULL |

Pilot_licenses Table:

| personID | license |
|---|---|
| p1 | jet |
| p10 | jet |
| p11 | jet |
| p11 | prop |
| p12 | prop |
| p13 | jet |
| p14 | jet |
| p15 | jet |
| p15 | prop |
| p15 | testing |
| p16 | jet |
| p17 | jet |
| p17 | prop |
| p18 | jet |
| p19 | jet |
| p2 | jet |
| p2 | prop |
| p21 | jet |
| p21 | prop |
| p22 | jet |
| p23 | jet |
| p24 | jet |
| p24 | prop |
| p24 | testing |
| p25 | jet |
| p26 | jet |
| p3 | jet |
| p4 | jet |
| p4 | prop |
| p5 | jet |
| p6 | jet |
| p6 | prop |
| p7 | jet |
| p8 | prop |
| p9 | jet |
| p9 | prop |
| p9 | testing |
| NULL | NULL |

Person Table:

| personID | first_name | last_name | locationID |
|---|---|---|---|
| p1 | Jeanne | Nelson | plane_1 |
| p10 | Lawrence | Morgan | plane_9 |
| p11 | Sandra | Cruz | plane_9 |
| p12 | Dan | Ball | plane_11 |
| p13 | Bryant | Figueroa | plane_2 |
| p14 | Dana | Perry | plane_2 |
| p15 | Matt | Hunt | plane_2 |
| p16 | Edna | Brown | plane_15 |
| p17 | Ruby | Burgess | plane_15 |
| p18 | Esther | Pittman | port_2 |
| p19 | Doug | Fowler | port_4 |
| p2 | Roxanne | Byrd | plane_1 |
| p21 | Mona | Harrison | port_4 |
| p22 | Arlene | Massey | port_2 |
| p23 | Judith | Patrick | port_3 |
| p24 | Reginald | Rhodes | plane_1 |
| p25 | Vincent | Garcia | plane_1 |
| p26 | Cheryl | Moore | plane_4 |
| p27 | Michael | Rivera | plane_7 |
| p28 | Luther | Matthews | plane_8 |
| p29 | Moses | Parks | plane_8 |
| p3 | Tanya | Nguyen | plane_4 |
| p30 | Ora | Steele | plane_9 |
| p31 | Antonio | Flores | plane_9 |
| p32 | Glenn | Ross | plane_11 |
| p33 | Irma | Thomas | plane_11 |
| p34 | Ann | Maldonado | plane_2 |
| p35 | Jeffrey | Cruz | plane_2 |
| p36 | Sonya | Price | plane_15 |
| p37 | Tracy | Hale | plane_15 |
| p38 | Albert | Simmons | port_1 |
| p39 | Karen | Terry | port_9 |
| p4 | Kendra | Jacobs | plane_4 |
| p40 | Glen | Kelley | plane_4 |
| p41 | Brooke | Little | port_4 |
| p42 | Daryl | Nguyen | port_3 |
| p43 | Judy | Willis | port_1 |
| p44 | Marco | Klein | port_2 |
| p45 | Angelica | Hampton | port_5 |
| p5 | Jeff | Burton | plane_4 |
| p6 | Randal | Parks | plane_7 |
| p7 | Sonya | Owens | plane_7 |
| p8 | Bennie | Palmer | plane_8 |
| p9 | Marlene | Warner | plane_8 |
| NULL | NULL | NULL | NULL |

**Query 19 Name:**

`flights_in_the_air()`

**Description:**

This view describes where flights that are currently airborne are located. We need to display what airports these flights are departing from, what airports they are arriving at, the number of flights that are flying between the departure and arrival airport, the list of those flights, the earliest and latest arrival times for the destinations and the list of planes (by the location id) flying these flights.

**Test Case:**

```
select * from flights_in_the_air;
```

**Expected Output:**

`Flights_in_the_air View:`

| departing_from | arriving_at | num_flights | flight_list | earliest_arrival | latest_arrival | airplane_list |
|---|---|---|---|---|---|---|
| DAL | HOU | 1 | SW_610 | 11:30:00 | 11:30:00 | plane_11 |
| LAX | ORD | 1 | SW_1776 | 14:00:00 | 14:00:00 | plane_9 |
| ORD | ATL | 1 | UN_523 | 11:00:00 | 11:00:00 | plane_8 |
| ORD | DCA | 1 | SP_1880 | 15:00:00 | 15:00:00 | plane_15 |

**Query 20 Name:**

`flights_on_the_ground()`

**Description:**

This view describes where flights that are currently on the ground are located. We need to display what airports these flights are departing from, how many flights are departing from each airport, the list of flights departing from each airport, the earliest and latest arrival time amongst all of these flights at each airport, and the list of planes (by their location id) that are departing from each airport.

**Test Case:**

```
select * from flights_on_the_ground;
```

**Expected Output:**

`Flights_on_the_ground View:`

| departing_from | num_flights | flight_list | earliest_arrival | latest_arrival | airplane_list |
|---|---|---|---|---|---|
| ORD | 1 | AM_1523 | 14:30:00 | 14:30:00 | plane_4 |
| ATL | 1 | DL_1174 | 08:00:00 | 08:00:00 | plane_1 |
| JFK | 1 | DL_1243 | 09:30:00 | 09:30:00 | plane_2 |
| SEA | 1 | UN_1899 | 09:30:00 | 09:30:00 | plane_7 |

**Query 21 Name:**

`people_in_the_air()`

**Description:**

This view describes where people who are currently airborne are located. We need to display what airports these people are departing from, what airports they are arriving at, the list of planes (by the location id) flying these people, the list of flights these people are on, the earliest and latest arrival times of these people, the number of these people that are pilots, the number of these people that are passengers, the number of people on the airplane, and the list of these people by their person id.

**Test Case:**

```
select * from people_in_the_air;
```

**Expected Output:**

`People_in_the_air View:`

| departing_from | arriving_at | num_airplanes | airplane_list | flight_list | earliest_arri... | latest_arrival | num_pilots | num_passengers | joint_pilots_passeng... | person_list |
|---|---|---|---|---|---|---|---|---|---|---|
| DAL | HOU | 1 | plane_11 | SW_610 | 11:30:00 | 11:30:00 | 1 | 2 | 3 | p12,p32,p33 |
| LAX | ORD | 1 | plane_9 | SW_1776 | 14:00:00 | 14:00:00 | 2 | 2 | 4 | p10,p11,p30,p31 |
| ORD | ATL | 1 | plane_8 | UN_523 | 11:00:00 | 11:00:00 | 2 | 2 | 4 | p28,p29,p8,p9 |
| ORD | DCA | 1 | plane_15 | SP_1880 | 15:00:00 | 15:00:00 | 2 | 2 | 4 | p16,p17,p36,p37 |

**Query 22 Name:**

`people_on_the_ground()`

**Description:**

This view describes where people who are currently on the ground are located. We need to display what airports these people are departing from by airport id, location id, and airport name, the city and state of these airports, the number of these people that are pilots, the number of these people that are passengers, the number people at the airport, and the list of these people by their person id.

**Test Case:**

```
select * from people_on_the_ground;
```

**Expected Output:**

`People_on_the_ground View:`

| departing_fr... | airport | airport_name | city | state | num_pilots | num_passengers | joint_pilots_passeng... | person_list |
|---|---|---|---|---|---|---|---|---|
| ATL | port_1 | Hartsfield-Jackson Atlanta International Airport | Atlanta | GA | 0 | 2 | 2 | p38,p43 |
| DCA | port_9 | Ronald Reagan Washington National Airport | Washington | DC | 0 | 1 | 1 | p39 |
| DEN | port_3 | Denver International Airport | Denver | CO | 2 | 2 | 3 | p20,p23,p42 |
| DFW | port_2 | Dallas-Fort Worth International Airport | Dallas | TX | 2 | 2 | 3 | p18,p22,p44 |
| LAX | port_5 | Los Angeles International Airport | Los Angeles | CA | 0 | 1 | 1 | p45 |
| ORD | port_4 | O_Hare International Airport | Chicago | IL | 2 | 2 | 3 | p19,p21,p41 |

**Query 23 Name:**

`route_summary()`

**Description:**

This view will give a summary of every route. This will include the routeID, the number of legs per route, the legs of the route in sequence, the total distance of the route, the number of flights on this route, the flightIDs of those flights, and the sequence of airports visited by the route.

**Test Case:**

```
select * from route_summary;
```

**Expected Output:**

`Route_summary View:`

| route | num_legs | leg_sequence | route_length | num_flights | flight_list | airport_sequence |
|---|---|---|---|---|---|---|
| circle_east_coast | 3 | leg_4,leg_20,leg_7 | 1800 | 2 | DL_3410,SP_1880 | ATL->ORD,ORD->DCA,DCA->ATL |
| circle_west_coast | 3 | leg_18,leg_10,leg_22 | 2800 | 2 | AM_1523,UN_717 | LAX->DFW,DFW->ORD,ORD->LAX |
| eastbound_north_milk_run | 3 | leg_24,leg_20,leg_8 | 2600 | 1 | UN_1899 | SEA->ORD,ORD->DCA,DCA->JFK |
| eastbound_north_nonstop | 1 | leg_23 | 2400 | 0 | NULL | SEA->JFK |
| eastbound_south_milk_run | 3 | leg_18,leg_9,leg_1 | 2600 | 0 | NULL | LAX->DFW,DFW->ATL,ATL->IAD |
| hub_xchg_southeast | 2 | leg_25,leg_4 | 1200 | 1 | UN_523 | ORD->ATL,ATL->ORD |
| hub_xchg_southwest | 2 | leg_22,leg_26 | 1600 | 1 | SW_1776 | ORD->LAX,LAX->ORD |
| local_texas | 2 | leg_12,leg_6 | 400 | 1 | SW_610 | IAH->DAL,DAL->HOU |
| northbound_east_coast | 1 | leg_3 | 800 | 1 | DL_1174 | ATL->JFK |
| northbound_west_coast | 1 | leg_19 | 1000 | 0 | NULL | LAX->SEA |
| southbound_midwest | 1 | leg_21 | 800 | 0 | NULL | ORD->DFW |
| westbound_north_milk_run | 3 | leg_16,leg_22,leg_19 | 2600 | 0 | NULL | JFK->ORD,ORD->LAX,LAX->SEA |
| westbound_north_nonstop | 1 | leg_17 | 2400 | 1 | DL_1243 | JFK->SEA |
| westbound_south_nonstop | 1 | leg_27 | 1600 | 0 | NULL | ATL->LAX |

**Query 24 Name:**

`alternative_airports()`

**Description:**

This view displays airports that share the same city and state. It should specify the city, state, the number of airports shared, and the lists of the airport codes and airport names that are shared.

**Test Case:**

```
select * from alternative_airports;
```

**Expected Output:**

`Alternative_airports View:`

| city | state | num_airports | airport_code_list | airport_name_list |
|---|---|---|---|---|
| Anchorage | AK | 2 | ANC,MRI | Ted Stevens Anchorage International Airport,Merrill Field |
| Chicago | IL | 2 | MDW,ORD | Chicago Midway International Airport,O_Hare International Airport |
| Dallas | TX | 2 | DAL,DFW | Dallas Love Field,Dallas-Fort Worth International Airport |
| Houston | TX | 2 | HOU,IAH | William P_Hobby Airport,George Bush Intercontinental Houston Airport |
| New York | NY | 2 | JFK,LGA | John F_Kennedy International Airport,LaGuardia Airport |
| Washington | DC | 2 | DCA,IAD | Ronald Reagan Washington National Airport,Washington Dulles International Airport |

**Query 25 Name:**

`simulation_cycle()`

**Description:**

This stored procedure executes the next step in the simulation cycle. The flight with the smallest next time in chronological order must be identified and selected.

If multiple flights have the same time, then flights that are landing should be preferred over flights that are taking off. Similarly, flights with the lowest identifier in alphabetical order should also be preferred.

If an airplane is in flight and waiting to land, then the flight should be allowed to land, passengers allowed to disembark, and the time advanced by one hour until the next takeoff to allow for preparations.

If an airplane is on the ground and waiting to takeoff, then the passengers should be allowed to board, and the time should be advanced to represent when the airplane will land at its next location based on the leg distance and airplane speed.

If an airplane is on the ground and has reached the end of its route, then the flight crew should be recycled to allow rest, and the flight itself should be retired from the system.

**Hint:** You may find it helpful to call some of the stored procedures you have already written.

**Test Case:**

```
call simulation_cycle();
```

**Expected Output:**

| flightID | routeID | support_airline | support_tail | progress | airplane_status | next_time |
|---|---|---|---|---|---|---|
| AM_1523 | circle_west_coast | American | n330ss | 2 | on_ground | 14:30:00 |
| DL_1174 | northbound_east_coast | Delta | n106js | 1 | in_flight | 12:00:00 |
| DL_1243 | westbound_north_nonstop | Delta | n110jn | 0 | on_ground | 09:30:00 |
| DL_3410 | circle_east_coast | NULL | NULL | NULL | NULL | NULL |
| SP_1880 | circle_east_coast | Spirit | n256ap | 2 | in_flight | 15:00:00 |
| SW_1776 | hub_xchg_southwest | Southwest | n401fj | 2 | in_flight | 14:00:00 |
| SW_610 | local_texas | Southwest | n118fm | 2 | in_flight | 11:30:00 |
| UN_1899 | eastbound_north_milk_run | United | n517ly | 0 | on_ground | 09:30:00 |
| UN_523 | hub_xchg_southeast | United | n620la | 1 | in_flight | 11:00:00 |
| UN_717 | circle_west_coast | NULL | NULL | NULL | NULL | NULL |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |