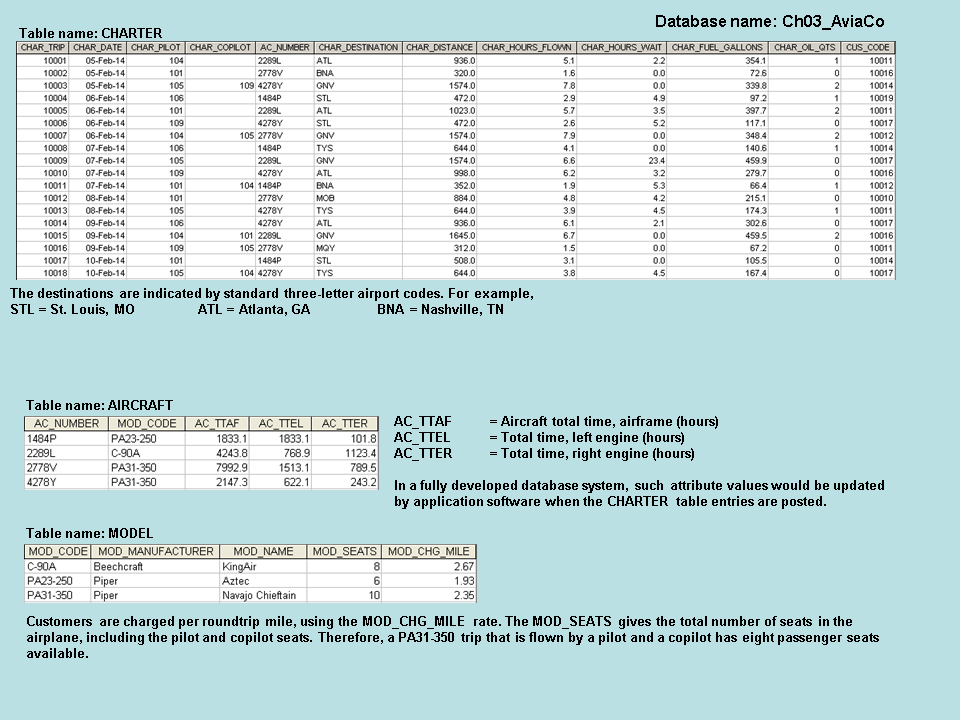
**HW 1**

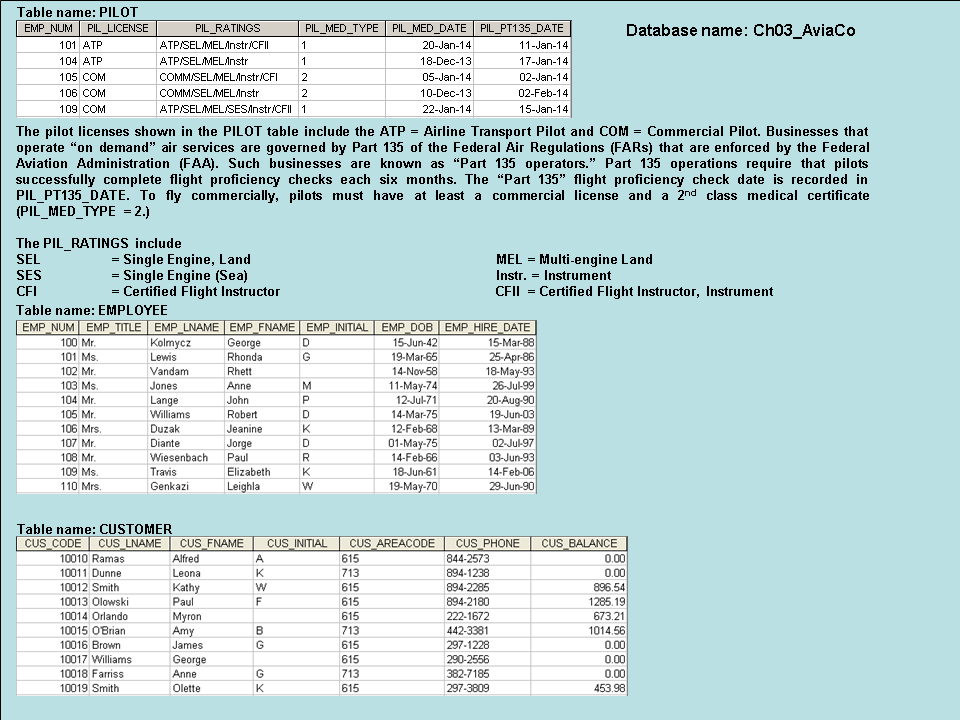
**Use the database shown in Figure P3.24 to answer Problems 24−33. AviaCo is an aircraft charter company that supplies on-demand charter flight services using a fleet of four aircraft. Aircrafts are identified by a unique registration number. Therefore, the aircraft registration number is an appropriate primary key for the AIRCRAFT table.**

**FIGURE P3.24 The Ch03\_AviaCo Database Tables (Part 1)**

****

**The nulls in the CHARTER table’s CHAR\_COPILOT column indicate that a copilot is not required for some charter trips or for some aircraft. Federal Aviation Administration (FAA) rules require a copilot on jet aircraft and on aircraft having a gross take-off weight over 12,500 pounds. None of the aircraft in the AIRCRAFT table are governed by this requirement; however, some customers may require the presence of a copilot for insurance reasons. All charter trips are recorded in the CHARTER table.**

**FIGURE P3.24 The Ch03\_AviaCo Database Tables (Part 2)**

****

1. **For each table, based on the current data (i.e., assuming no new data will be added), identify:**
   1. **The primary key**
   2. **A superkey**
   3. **A candidate key** (you can suggest a new attribute if a candidate key is not readily available)
   4. **The foreign key(s)**
   5. **A secondary key**

**Answer 24.**

1. **Primary Key**

|  |  |
| --- | --- |
| **Table Name** | **Primary Key** |
| **AIRCRAFT** | **AC\_NUMBER** |
| **CHARTER** | **CHAR\_TRIP** |
| **CUSTOMER** | **CUS\_CODE** |
| **EMPLOYEE** | **EMP\_NUM** |
| **MODEL** | **MOD\_CODE** |
| **PILOT** | **EMP\_NUM** |

1. **Super Key**

|  |  |
| --- | --- |
| **Table Name** | **Super Key** |
| **AIRCRAFT** | **AC\_NUMBER + MOD\_CODE** |
| **CHARTER** | **CHAR\_TRIP + CHAR\_DATE** |
| **CUSTOMER** | **CUS\_CODE + CUS\_LNAME** |
| **EMPLOYEE** | **EMP\_NUM + EMP\_DOB** |
| **MODEL** | **MOD\_CODE + MOD\_NAME** |
| **PILOT** | **EMP\_NUM + PIL\_LICENSE** |

1. **Candidate Key**

|  |  |
| --- | --- |
| **Table Name** | **Candidate Key** |
| **AIRCRAFT** | **AC\_NUMBER** |
| **CHARTER** | **CHAR\_TRIP** |
| **CUSTOMER** | **CUS\_LNAME + CUS\_FNAME + CUS\_INITIAL + CUS\_PHONE** |
| **EMPLOYEE** | **EMP\_LNAME + EMP\_FNAME + EMP\_INITIAL + EMP\_DOB** |
| **MODEL** | **MOD\_CODE** |
| **PILOT** | **EMP\_NUM** |

1. **Foreign Key**

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Foreign Key (s)** | **Comments** |
| **AIRCRAFT** | **MOD\_CODE** | **References MOD\_CODE of MODEL table** |
| **CHARTER** | **CHAR\_PILOT** | **References EMP\_NUM of PILOT table** |
| **CHARTER** | **CHAR\_COPILOT** | **References EMP\_NUM of PILOT table** |
| **CHARTER** | **AC\_NUMBER** | **References AC\_NUMBER of AIRCRAFT table** |
| **CHARTER** | **CUS\_CODE** | **References CUS\_CODE of CUSTOMER table** |
| **CUSTOMER** | **None** |  |
| **EMPLOYEE** | **None** |  |
| **MODEL** | **None** |  |
| **PILOT** | **EMP\_NUM** | **References EMP\_NUM of EMPLOYEE table** |

1. **Secondary Key**

|  |  |
| --- | --- |
| **Table Name** | **Secondary Key** |
| **AIRCRAFT** | **MOD\_NAME** |
| **CHARTER** | **CHAR\_DATE + AC\_NUMBER + CHAR\_DESTINATION** |
| **CUSTOMER** | **CUS\_LNAME + CUS\_FNAME + CUS\_PHONE** |
| **EMPLOYEE** | **EMP\_LNAME + EMP\_FNAME + EMP\_DOB** |
| **MODEL** | **MOD\_MANUFACTURER + MOD\_NAME** |
| **PILOT** | **PIL\_LICENSE + PIL\_MED\_DATE** |

1. **Create the ERD in Visio and copy it here. (*Hint*: Look at the table contents. You will discover that an AIRCRAFT can fly many CHARTER trips but that each CHARTER trip is flown by one AIRCRAFT. Similarly, you will discover that a MODEL references many AIRCRAFT but that each AIRCRAFT references a single MODEL, etc.)**

In each entity, please at least include the PK and FK attributes, and the required attributes. If you are making any assumptions about an optional (i.e., a minimum cardinality of 0) or mandatory relationship (i.e., a minimum cardinality of 1), please state the assumptions below.

Note that all pilots are employees, but not all employees are pilots – some are mechanics, accountants, and so on (i.e., EMPLOYEE is the “parent” of PILOT). The relationship between PILOT and EMPLOYEE is read from the “parent” entity to the related entity.

Diagram

Description automatically generated

1. **Create the relational diagram in Access by connecting all tables in the relationship view and copy it here. Note that referential integrity should be enforced on all relationships.**

**Diagram

Description automatically generated**

1. **Modify the ERD you created in Problem 25 to eliminate the problems created by the use of synonyms and copied the revised ERD here. (*Hint:* Modify the CHARTER table structure by eliminating the CHAR\_PILOT and CHAR\_COPILOT attributes; then create a composite table named CREW to link the CHARTER and EMPLOYEE tables. Some crewmembers, such as flight attendants, may not be pilots. That’s why the EMPLOYEE table enters into this relationship.)**

**Diagram

Description automatically generated**

1. **Draw the relational diagram for the design revised in problem 27 in Access and copy it here.**

**Diagram

Description automatically generated**

**You are interested in seeing data on charters flown by either Mr. Robert Williams (employee number 105) or Ms. Elizabeth Travis (employee number 109) as pilot or copilot, but not charters flown by both of them. Complete problems 29 – 31 to find these data.**

**NOTE: For Probs 27 and Prob 28, the access database attached is HW1\_Mukherjee\_Santanu- Prob 27-28.mdb**

1. **Use Access query to create the table that would result from applying the SELECT and PROJECT relational operators to the CHARTER table to return only the CHAR\_TRIP, CHAR\_PILOT, and CHAR\_COPILOT attributes for charters flown by either employee 105 or employee 109. Name your query as Prob\_29.**

|  |  |  |
| --- | --- | --- |
| **CHAR\_TRIP** | **CHAR\_PILOT** | **CHAR\_COPILOT** |
| 10003 | 105 | 109 |
| 10006 | 109 |  |
| 10007 | 104 | 105 |
| 10009 | 105 |  |
| 10010 | 109 |  |
| 10013 | 105 |  |
| 10016 | 109 | 105 |
| 10018 | 105 | 104 |

1. **Use Access query to create the table that would result from applying the SELECT and PROJECT relational operators to the CHARTER table to return only the CHAR\_TRIP, CHAR\_PILOT, and CHAR\_COPILOT attributes for charters flown by both employee 105 and employee 109. Name your query as Prob\_30.**

|  |  |  |
| --- | --- | --- |
| **CHAR\_TRIP** | **CHAR\_PILOT** | **CHAR\_COPILOT** |
| 10003 | 105 | 109 |
| 10016 | 109 | 105 |

1. **Use Access query to create the table that would result from applying a DIFFERENCE relational operator of your result from problem 29 to your result from problem 30. Name your query as Prob\_31.**

|  |  |  |
| --- | --- | --- |
| **CHAR\_TRIP** | **CHAR\_PILOT** | **CHAR\_COPILOT** |
| 10006 | 109 |  |
| 10007 | 104 | 105 |
| 10009 | 105 |  |
| 10010 | 109 |  |
| 10013 | 105 |  |
| 10018 | 105 | 104 |

1. **Use Access query to create a table that shows all employees who are not a pilot (using the Access design view, not SQL view). Name your query as Prob\_32.**
2. **The CHARTER table shows some flights operated by a pilot and a copilot. Use Access query to create a table that shows pilots who have never worked together (using the Access design view, not SQL view). Name your query as Prob\_33.**

**NOTE: For Probs 25, 26, 29-32, the access database attached is HW1\_Mukherjee\_Santanu.mdb**

**Could not do Prob 33 using Access Design View, but could have done it in SQL**

**Attachment of HW1\_Mukherjee\_Santanu.mdb**

**Graphical user interface, application

Description automatically generated**

**Prob 29**

**Graphical user interface, table

Description automatically generated with medium confidence**

**Prob 30**

**Graphical user interface, application, table

Description automatically generated**

**Prob 31**

**Graphical user interface, application, table

Description automatically generated**

**Prob 32**

**Graphical user interface, table

Description automatically generated**