

Teaching Case

Knowledge Creation through User-Guided Data Mining: A Database Case

David M. Steiger
Maine Business School
University of Maine
Orono, Maine 04469-5723 USA
dsteiger@maine.edu

ABSTRACT

This case focuses on learning by applying the four integrating mechanisms of Nonaka's knowledge creation theory: socialization, externalization, combination and internalization. In general, such knowledge creation and internalization (i.e., learning) is critical to database students since they will be expected to apply their specialized database knowledge to a wide variety of business problems, incorporating the various concepts of multiple business disciplines. The case, presented below, depicts a four-phase consulting assignment for each two-student team. In Phase 1, the teams design and implement a reservations database for a small corporate airline, basing their design on a written list of requirements. In Phase 2, each team uses the resulting database to address questions (queries) posed by key managers in the client firm. In Phase 3, each team develops operational and profit improvement recommendations for consideration by the firm's top management. And finally, Phase 4 requires each team to present their recommendations to the client firm's Board of Directors (the rest of the database class), answering any questions and concerns raised by this august board. Taken together, the four phases constitute knowledge creation through user-guided data mining. Since the queries require a variety of functions, subqueries, and cascading views, this case is most suitable as an integrative term project in either an upper-division undergraduate or a graduate level database course.

Keywords: Knowledge Creation, Database Project, User-guided Data Mining, Profit Improvement

1. PROJECT SUMMARY

Terra Firma, Inc. (TFI) is a growing management consulting firm headquartered in Bangor, Maine. TFI has major, on-going consulting assignments in several U.S. cities, and is currently expanding into other U.S. regions, as well as Western Europe. To transport its consultants to and from their assignments, the firm has purchased a small fleet of seven (previously owned) corporate jets. TFI now needs your assistance in building a reservation, scheduling, and reporting database system for its airline operations. It should be noted that one of the fringe benefits of Terra Firma employment is that employees are provided free transportation to and from the vacation locations of their choice, a benefit that most employees frequently use.

2. PROJECT BACKGROUND AND REQUIREMENTS

The TFI Reservation System must be designed to satisfy several requirements described below. (Note: the attribute

formats are given in parenthesis within the text below; e.g., D4 indicates a four digit numeric field, C5 indicates a five character alphanumeric field.)

2.1 Aircraft Information

The airline has several types of aircraft; each type of aircraft, uniquely identified by a type code (C5), has a specified passenger capacity (D3), a flight range (D4), a type description (C25), and the maximum number of flight hours (D3) between scheduled maintenance appointments (an aircraft is out of service for one week during its scheduled maintenance). There are one or more aircraft of each type in the airline's fleet. Each specific aircraft has a unique serial number (C6), a type code, the date of its most recent maintenance (C8), and its current fair market value (D7). Each specific aircraft can fly at most one round trip each day.

2.2 Pilot Information

Pilots, uniquely identified by a pilot ID (C5), are certified to fly only certain type(s) of aircraft. Pilot information includes

the pilot's name (C8), address (C15), city (C15), state (C2), zip code (C5), phone number (C13), and the type codes of all aircraft s/he is certified to fly. Each pilot flies at most one round trip/day and at most 30 hours/week (air time), and is paid an annual salary of \$80,000 plus 28% benefits; pilots are non-union employees.

2.3 Flight Information

Each *flight*, identified by a unique flight number (C3), has an origination city code (C3) and a destination city code (C3) (there are currently no intermediate stops), estimated time of departure (D4) from the origination city (ETD), estimated time of arrival (D4) at the destination city (ETA), flight distance from origin to destination (in miles), a round trip id (C3) (a round trip includes the outbound leg and the return flight), and the commercial price (D5) of a 1st class round trip ticket to this destination. All times are recorded using a military designation (i.e., using a 24 hour clock), where 9:30 a.m. is 0930, 1:15 p.m. is 1315, etc.; both the ETD and ETA are specified in terms of the local standard time of the flight's origination city. Further, each origination/destination city has a city code (C3), the full name of the city (C10), state/country code, and a brief description of the runway facilities (C28).

2.4 Scheduled Flight Information

A *scheduled flight* is a flight that has been scheduled for a specific date, and includes the flight number (C3), flight date (C8), aircraft serial number, and a pilot ID, in addition to the flight information listed above; a scheduled flight is uniquely identified by the combination of a flight number and date.

2.5 Passenger Information

Passengers can make reservations on any scheduled flights; however, they are not assigned specific seats. For purposes of this database, we will assume that passengers are uniquely identified by name (C10); additional passenger information includes his/her department (C2), job grade (C2), and phone number (C13). When passengers book reservations on a scheduled flight, they must provide a project ID (C4) to which the fare will be charged. Each passenger can take at most one round trip/day.

2.6 Commercial Bookings

If TFI scheduled flights cannot accommodate an employee's travel plans, then Terra Firma will purchase him/her a 1st class round trip ticket on a commercial flight, recording the employee's name, flight date, origin city code, destination city code, flight distance from origin to destination (in miles), commercial fare, and the project ID to which the fare will be charged.

2.7 Break-Even Calculations

Management has determined that the profit break even point on any scheduled round trip flown by TFI occurs when the total number of passengers (summed over both legs of the round trip) is half of the aircraft's passenger capacity. For example, if an aircraft's passenger capacity is 5, then a flight flown by that aircraft will be profitable if there are 3 passengers on the outbound leg and none on the return leg, or 2 on the outbound leg and 1 on the return leg; likewise, the round trip will lose money if there are 2 or fewer total

passengers. The specific amount of profit/loss for a round trip is calculated as: the total number of passengers (summed over both legs of the round trip) less half of the aircraft's capacity, multiplied by half of the 1st class round trip commercial fare for that destination.

Hint: One way to calculate the profit/loss of a scheduled round trip is to first, create a view that includes Flt#, FltDate, and PassngrCount, using an outer join on the Reservation table join conditions to include any scheduled flights that have zero passengers. Then, use this view to create a second view with Flt#, FltDate, City, RoundtripID, AircraftID, PilotID, RTCommmlAirfare, PassngrCount, and Profit [= (PassngrCount - Capacity/4) * RTCommmlAirfare/2]. Finally, create a view that calculates the round trip profits by summing the outbound and return trip profits and passenger counts for each round trip on each date.

2.8 Special Calculations

For the purpose of aircraft maintenance, the system must calculate the hours flown during each scheduled round trip (including the return trips to Bangor) based on the estimated time of departure (ETD) and estimated time of arrival (ETA). Thus, for each *round trip*, we must calculate and list the estimated flying time (in hours, rounded to the nearest 2 decimal places) based on the ETD and ETA. Since both the ETD and ETA for each flight are specified in terms of the standard time in the origination city, you don't need to adjust for time zone changes. However, since ETDs and ETAs are expressed in military time, you can't simply subtract the ETD from the ETA to get the flying time; e.g., 1500-1345 = 155 which is not 1 hr and 15 minutes. (Hint: you might want to convert each time to the number of hours, and fractions of hours, since midnight by using two SQL number functions: 1) FLOOR(number) which returns the value of a number (or arithmetic expression) in the argument, rounded *down* to the next integer; e.g., FLOOR(1545/100) = 15, and 2) MOD(number, divisor) which returns the remainder (modulus) for a number and its divisor; e.g., MOD(1545,100) = 45 minutes past the hour. Now if you divided the minutes by 60, you would convert the minutes into a decimal fraction of an hour which you could then add to the hour. For example, to convert the military time of 1545 to a decimal equivalent of hours since midnight, the calculation would be as follows: FLOOR(1545/100) + MOD(1545,100)/60 = 15 + 45/60 = 15.75. This could then be subtracted from the corresponding ETA to generate the estimated flying time.)

2.9 First Deliverable

Your first assignment in this case is to generate a set of tables and attributes (in third normal form), including for each proposed table, the table name, a list of attributes and their formats, underlining the primary key attribute(s) and noting any foreign key(s) and the referenced table(s). Also required is a sample tuple for each table (make up some data in the appropriate format). Note that TFI management is not interested in the normalization process, just the results.

3. TERRA FIRMA REQUIRED INFORMATION

Terra Firma's Manager of Aviation Operations, Mr. U. Tossit Farnwide, recently read an article expressing the

benefits of a new database technology called “user-guided data mining”, which is the process of extracting previously unknown (but suspected), valid and actionable information from large databases, based on hypotheses provided by knowledgeable users. Mr. Farnwide is interested in the possibilities of applying this technology to the new TFI Reservation and Scheduling System once it has been created and thoroughly tested.

In preparation for this task, Mr. Farnwide assembled the top executives in the TFI Airline Division to determine critical information required to improve profits and operations in the Division. More specifically, they composed a set of questions (queries) that would be necessary in this task. Appendix 1 specifies a set of preliminary views to familiarize you with the TFI tables. Appendix 2 includes the first half of the executive queries, while Appendix 3 includes the second half of executive queries. In addition, TFI’s analysts created a “representative week” of consultant air travel requirements based on existing and expected long-term consulting contracts. This representative travel week will be provided by your instructor as part of an SQL script file that creates the case tables and instantiates each table with the appropriate TFI data.

Thus, your second deliverable in this project requires you first, to execute the script file, creating the appropriate tables in the database and instantiating each table with TFI’s data; second, to familiarize yourself with the table names, attribute names, primary keys, etc.; and third, to design, write, debug, and print out the views, queries, and results required by TFI’s executives.

4. OPERATIONAL/PROFIT IMPROVEMENT RECOMMENDATIONS

As your third deliverable, TFI has requested you to utilize the new Reservation and Scheduling System to recommend ways to improve the operations and profits of its TFI Air Division; i.e., 1) recommend specific ways to eliminate violations of government and corporate rules and regulations, and 2) recommend specific ways to improve the overall profits through resource re-allocation, and/or by eliminating/modifying/adding specific flights or scheduled flights to improve profits, and/or by recommending other ways to improve profits or operations.

TFI is willing to relax some of its current flight and scheduling restrictions if there is sufficient economic incentive to do so. Several potential policy relaxations are provided below for incorporation into your recommendations.

4.1 Layovers

Currently each round trip is completed within the same calendar day, if physically possible, or on succeeding calendar days, if not. However, Terra Firma’s management would consider flight layovers in distant cities if the aircraft and crew are not scheduled for other flights during the extended round trip and if such layovers could be shown to be more profitable than flying the aircraft and crew home to Bangor each day; layover costs are estimated at \$2000/day, regardless of the location. Any potential layover recommendations should include the incremental

profitability associated with the layover, including the layover costs.

4.2 Intermediate Stops

Likewise, all TFI flights are currently direct, non-stop flights. However, management would consider flights with, at most, one intermediate stop if such flights covered the \$500 incremental cost of the intermediate stop.

4.3 Cancelling Scheduled Flights

Finally, TFI management would consider any recommendations for canceling a scheduled round trip, but you must cancel both legs of the round trip. For example, if you have a full flight from Bangor to Dallas and an empty flight back from Dallas to Bangor, you cannot simply cancel the flight back – you must cancel both outbound and return flights. Further, if you cancel a scheduled round trip, and there are one or more passengers with reservations on either leg of that round trip, you must buy each of those passengers a first class round trip commercial airfare ticket; thus, canceling one roundtrip may affect the profitability of other roundtrips to the same destination.

4.4 Pro-Forma Income Statement

TFI, Inc., originally created the TFI Airline Department (TFI Air) as a profit center, setting the ticket transfer price equal to the first class commercial airfare. That is, all air travel requests are funneled through TFI Air, and TFI Air bills the appropriate departments for the price of the round trip first class commercial airfare. TFI Air then tries to maximize profits based on the revenue generated from that transfer price and the expenses associated with flying.

Currently, TFI Air has both fixed costs and variable costs, both of which must be included in the profit calculations. Fixed costs consist of two components: aircraft depreciation (assume 5 year straight line depreciation based on current fair market value with no salvage value; currently depreciation is \$879,000/year), and pilot’s salary (\$80,000 + 28% benefits, or \$102,400/year for each pilot).

Variable costs are lumped together and are estimated (for a round trip) to be half of the aircraft’s passenger capacity. The pro forma Income Statement, developed in QB#9, is a good way to keep track of your profit improvements. That is, as you make a recommendation, update the appropriate TFI table to reflect the implementation of your recommendation, and then list the data in the Income Statement to see what effect it has on the earnings before interest and taxes (EBIT). TFI’s latest Pro Forma Income Statement is provided in Figure 1 below for your reference.

4.5 Forecasted Growth

For the purpose of this case, assume a growth rate at TFI Air of 5%/year, all of it in new business, with half of the new business in cities which TFI currently has contracts, and the other half in cities which TFI currently has no business.

4.6 Recommendations

To generate recommendations, you should read and analyze the results generated in response to the questions (queries) that TFI management posed.

Revenues	
TFI reservation revenues	\$ 3,879,252
Comm'l reservations revenues	6,012,552
Total TFI Revenues	\$ 9,891,804
Expenses	
TFI flight expense	\$ 2,215,395
Commercial ticket expense	6,012,552
Salary expense	716,800
Depreciation expense	879,000
Total Expenses	\$9,823,747
Earnings before Interest & Taxes	\$ 68,057
Gains from Sales of Assets	\$ 0

Figure 1. Pro Forma Income Statement for TFI

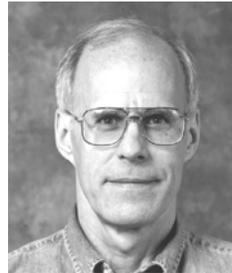
In preparation for a preliminary interview with Mr. Farnwide, your consulting group should generate several recommendations to eliminate operational violations and/or improve profits at TFI. Each recommendation should be specified in a 1-page summary that includes the following items: 1) a clearly stated, one sentence statement of the recommendation, 2) a summary table (view) of relevant information that supports your recommendation (e.g., before and after views that depict the operations/profits 'before' the implementation of your recommendation and 'after' the implementation), 3) an estimate of the *defendable* benefits (operational and/or dollar) realizable by implementing your recommendation, 4) a brief explanation (25 words or less) addressing the validity of each recommendation based on basic business principles of marketing, management, finance, and/or accounting, and 5) a list of assumptions that you made in determining the benefits/profits improvements of implementing your recommendation (e.g., there is sufficient capital to purchase an additional aircraft, and there is a pilot that is both qualified to fly the purchased aircraft and won't be overbooked as the result of the additional airtime). Note:

TFI executives are not interested in your brilliant SQL, just your results.

It should be noted that Mr. Farnwide takes a dim view of consultant recommendations for which there is no firm basis. That is, he requests that the benefits, assumptions and calculations for any recommendation be based on observed facts (preferably on the representative data in the TFI database), or at least on good business sense. Thus, you would be well advised to check with him or one of his assistants concerning these critical assumptions/calculations to see if relevant facts/business assumptions are used.

AUTHOR BIOGRAPHY

David M. Steiger is an Associate Professor of Management Information Systems at the University of Maine. He received his B.S. in Electrical Engineering and M.B.A. from the University of Texas at Austin, and a Ph.D. in Management Science/Information Systems from Oklahoma State University. Between his M.B.A. and Ph.D. degrees, he spent 15 years in various analysis and managerial positions in industry.



Professor Steiger's research interests include knowledge management, decision support systems, model analysis systems, and inductive artificial intelligence technologies. His articles have appeared in *Information Systems Research*, *Journal of MIS*, *Management Science*, *Decision Support Systems*, *Interfaces*, *INFORMS Journal On Computing*, *European Journal of Operational Research*, *Journal of Information Systems Education*, *Knowledge Management Research and Practice*, and *Journal of Knowledge Management*.

APPENDIX 1.
TFI CASE QUERIES: PRELIMINARY VIEWS

The purpose of this set of preliminary views is to familiarize you with the TFI tables, to demonstrate how to join them (especially those with composite keys), and to demonstrate several calculations with times and dates. For each query list (and clearly label with the appropriate query number) the **final query output only**. Then in an appendix, list (and clearly label with the appropriate query number) the views and queries that were used to generate the final output. TFI accountants have laboriously calculated and provided some numbers that you can use to check your output – these check-sums are provided after each query and usually consist of the first few rows of the output you should expect to get.

1. Create a view, named Flights_Plus, that contains all the information in the Flights table plus the airtime for each flight. The airtime is given by the following formula:

$(\text{FLOOR}(\text{ETA}/100)+(\text{MOD}(\text{ETA},100)/60)) - \text{FLOOR}(\text{ETD}/100)+(\text{MOD}(\text{ETD},100)/60))$

Hint: All required information is in the Flights table.

FLT#	ORIGIN	ETD	DEST	ETA	RTID	AIRTME
130	BGR	600	DFW	1045	7	4.75
135	DFW	1800	BGR	2245	7	4.75

2. Create a view, named Passngr_Count, that includes the flight #, flight date, and number of passengers holding reservations for each scheduled flight. Hint: Join the Sched_Flights table with the Reservations table using two **outer joins** on the Reservations table; using the outer joins will include those scheduled flights that have no passengers. Include a COUNT(DISTINCT PASSNGR) in your SELECT clause to count the number of different passengers.

FLT#	FLTDATE	PASSNGRS
130	13-NOV-06	3
130	17-NOV-06	0

3. Create a view, named Sched_Flights_Profit, that includes all the information in the Sched_Flights table plus the destination (not Bangor), the round trip ID, the round trip commercial airfare, the passenger capacity of the plane assigned to the scheduled flight, the number of passengers with reservations, and the profitability of the scheduled flight. The profit formula for a scheduled flight is:

$(\text{Passngrs} - (\text{Capacity}/4)) * (\text{RtFare}/2)$

Sort your results in ascending order by flight date and round trip ID. Hint: Join the Sched_Flights table and the Passngr_Count view created above (this requires two join conditions – on flt# and fltdate) and the Flight table (joined on Flt#) and the Aircraft table (joined on Aircraftid) and the Plane_Type table (joined on TypeCode) and the Cities table (using the following join conditions:

AND (CITY = DEST OR CITY = ORIGIN)
AND CITY <> 'BGR'

FLT	FLTDATE	CITY	RT	PLANE	PILOT	FARE	CAP	PASS	PROFIT
130	13-NOV-06	DFW	7	101010	P1234	1885	7	3	1178.13
135	13-NOV-06	DFW	7	101010	P1234	1885	7	0	-1649.38

4. (QA#1) A pilot is 'overbooked' if s/he is assigned to fly more than two scheduled flights (i.e., more than one roundtrip) on any day. Create a view, named Overbooked_Pilots, that contains the pilot's ID and flight dates of all 'overbooked' pilots, and the number of flights s/he is scheduled for on that date. Hint: Use the Sched_Flights table with a HAVING COUNT(*) >2 or a HAVING COUNT(DISTINCT FLT#) >2 clause.

PILOTID	FLTDATE	NUMFLTS
P9996	13-NOV-06	4

5. (QA#2) An aircraft is 'overbooked' if it is assigned to fly more than 2 scheduled flights (i.e., more than 1 round trip) on a single day. Create a view, named Overbooked_Planes, that contains the aircraft ID, and flight date of all 'overbooked' aircraft. Hint: Use the Sched_Flights table.

AIRCRAFTID	FLTDATE	NUMFLTS
101010	17-NOV-06	4
101010	13-NOV-06	4

6. (QA#5, QB#7) Create a view, named Idle_Pilots, that for every date that TFI has scheduled flights, includes the date, the PilotID, pilot's phone number, and the total scheduled weekly airtime for those pilots that are not currently assigned to a scheduled flight on that date. Sort your output by date and piloted. Hint: First create a view called Flt_Dates that has the seven different dates of current scheduled flights (use a SELECT DISTINCT...). Then, join the Flt_Dates view with the Pilots table (without a join condition) and a temporary table joined on pilotid (the temporary table contains the piloted and the total airtimes for each pilot, uses the Flights_Plus view and the Sched_Flights table, with a GROUP BY PILOTID clause). To exclude pilots already scheduled for a flight, use a PILOTID NOT IN clause in the outer query with a correlated subquery; the subquery should use the Sched_Flights table with the correlation condition based on the flight date. The first 25 rows of the Idle_Pilots view are given below:

FLTDATE	PILOTID	PHONE	WKLYAIRTIME
13-NOV-06	P2334	(207)583-1498	31.5
13-NOV-06	P5432	(207)583-3434	10

7. (QA#5, QA#8) Create a view, named Idle_Aircraft, that for every date that TFI has scheduled flights, includes the date and the AircraftIDs, types, capacities, and ranges of aircraft that are not currently assigned to a scheduled flight. Sort your results in ascending order by fltdate and aircraftid. Hint: First create a view named Dates_Aircraft that consists of all possible combinations of flight dates and aircraft IDs (hint – join the Flt_Dates view created above with the Aircraft table without a join condition, creating a 49 row view). Finally, to exclude aircraft already assigned to a scheduled flight, use an AIRCRAFTID NOT IN clause with a correlated subquery to get the required info; the subquery should use the Sched_Flights table with the correlation condition based on the flight date. The first 25 rows of the Idle_Aircraft view are given below:

FLTDATE	AIRCRAFTID	TYPE	CAP	RANGE
13-NOV-06	123456	CMUS	5	1500
13-NOV-06	235440	CMUS	5	1500

8. (QB#1) An 'overbooked passenger' is one who has reservations on more than 2 scheduled flights (i.e., more than one round trip) on a single day. Create a view named 'Overbooked_Passengers' that includes, for every overbooked passenger, the passenger's name, flight date, and the number of different flights the passenger is scheduled to fly on the date. Hint: Use the Reservations table with a HAVING clause to keep this information only on passengers scheduled for more than 2 different flights.

PASSNGR	FLTDATE	NUMFLTS
CAROLE	13-NOV-06	3
GERRINGER	13-NOV-06	4

9. (QB#2) A 'pilot-aircraft mismatch' occurs when a pilot is scheduled to fly an aircraft s/he is not certified to fly. Create a view, named Mismatches, that joins the Sched_Flights and Aircraft tables in the outer query to include the flight #, flight date, scheduled aircraft, typecode, and scheduled pilot; include a TYPECODE NOT IN clause and a correlated subquery to find the pilots not certified to fly the scheduled type of aircraft. The subquery should use the Certification table with a correlation condition based on the PilotID to find the pilots who are certified to fly the scheduled type of aircraft.

FLT#	FLTDATE	AIRCRAFTID	TYPECODE	PILOTID
170	15-NOV-06	101010	GIV	P6546
175	15-NOV-06	101010	GIV	P6546

10. (QB#3) Create a view named WklyAirtime that contains the aircraft ID and total airtime each aircraft is scheduled to fly during the 'representative' week. Hint: join the Sched_Flights table with the Flights_Plus view you created earlier, and SUM the airtimes.

AIRCRAFTID	WKLYHRS
101010	47.5
123456	3.5

11. (QB#6) Create a view, named Outbound_Passngrs, that contains the flight #, flight date, passenger name, roundtrip ID, and destination for all outbound passengers; i.e., passengers whose origin is 'BGR'. Hint: join Flights and Reservations on Flt#. Sort your results in ascending order by fltdate and round trip ID. (The first 24 rows of output are given below.)

FLT#	FLTDATE	PASSNGR	RTID	DEST
138	13-NOV-06	BILBRO	1	NYL
138	13-NOV-06	CAROLE	1	NYL

12. (QB#6) Create a view, named `Inbound_Passngrs`, that contains the flight #, flight date, passenger name, roundtrip ID, and destination for all inbound passengers; i.e., passengers whose destination is 'BGR'. Sort your results in ascending order by `fltdate` and round trip ID. Hint: join the `Flights` and `Reservations` tables (on `Flt#`).

FLT#	FLTDATE	PASSNGR	RTID	DEST
211	13-NOV-06	BILBRO	1	BGR
211	13-NOV-06	CAROLE	1	BGR

13. (QB#8) Create a view, called `All_Passngrs`, that contains the flight date, origin, destination and passenger name for all passengers flying either TFI or commercial flights or both. Hint: Use a UNION to combine the TFI passengers (joining the `Reservations` and `Flights` tables) and the commercial air passengers (from the `Fly_Comml` table).

FLTDATE	ORIGIN	DEST	PASSNGR
13-NOV-06	ATL	BGR	LING
13-NOV-06	ATL	BGR	ZING
13-NOV-06	BGR	ATL	LING
13-NOV-06	BGR	ATL	ZING

14. (QB#9) Create a view named `TFI_INCOME` that includes the total revenue generated from reservations on TFI scheduled flights (equals $0.5 * \text{round trip commercial airfare} * \text{the number of passengers on the scheduled flight}$ summed over all TFI scheduled flights), the total variable costs of TFI scheduled flights (`TFIEXP`) (equals $0.25 * \text{Capacity}$ summed over all scheduled flights in the `Sched_Flights_Profit` view), and the total pilot expense (`PILOT`) (equals the number of different pilots multiplied by $80,000 * 1.28$ to include both salary and benefits. (Hint: all required information for this query is contained in the `Sched_Flights_Profit` view.)

REVT	TFIEXP	PILOT
4173312	2558465	716800

APPENDIX 2.
TERRA FIRMA AIRLINES - QUERIES A

After creating the database and loading the test data, TFI requests that you use the data to generate the appropriate responses to the Executive Committee's fifteen questions (see below). Specifically, for each query list (and clearly label with the appropriate query number) the **final query output only**. Then in an appendix, list (and clearly label with the appropriate query number) the views and queries that were used to generate the final output. TFI accountants have laboriously calculated and provided some numbers that you can use to check your output – these check-sums are provided after most queries and usually consist of the first few rows of the output you should expect to get; for those queries without check-sums, you'll need to do some checking on your own to validate your results. Unless otherwise specified in the query, you should round all dollar amounts to the nearest penny, and all percentages to the nearest tenth of a percent. Also, answer any questions below by filling in the blanks provided, and handing in these pages as the cover sheets of your output packet. That is, each team should hand in one answer sheet, one set of final query outputs, and one appendix.

1. Overbooked pilots – note that a pilot is 'overbooked' if s/he is scheduled for more than 1 round trips in a single day. Create a view that, for each overbooked pilot, lists the pilot ID, date on which s/he is overbooked, and for each scheduled flight for the overbooked pilot on that date, list the round trip ID, origin city, ETD, destination city, and ETA. Sort your results in ascending order by pilot ID and round trip ID. (Hint: join the Sched_Flights table with the Overbooked_Pilots view (see PV#4) (on FltDate and PilotID) and Flights table (on Flt#), to provide the requested info.)

PILOTID	FLTDATE	RTID	ORIGIN	ETD	DEST	ETA
P9996	13-NOV-06	1	BGR	630	NYL	745
P9996	13-NOV-06	8	DFW	1300	BGR	1745

2. Overbooked planes: Create a view that, for each aircraft scheduled for more than one round trip on a single day, lists the aircraft serial number, date on which that aircraft is scheduled for more than one round trip, and for each scheduled flight for the overbooked aircraft on that date, list the round trip ID, origin city code, ETD, destination city code, and ETA. Sort your results in ascending order by round trip ID and ETD. (Note: if there are no overbooked planes, your output should be 'no rows selected'.) Hint: join the Overbooked_Planes view created in PV#5 with the Sched_Flights table (on FltDate and AircraftID) and the Flights table (on Flt#).

AIRCRAFTID	FLTDATE	RTID	ORIGIN	ETD	DEST	ETA
101010	13-NOV-06	7	BGR	600	DFW	1045
101010	13-NOV-06	7	DFW	1800	BGR	2245

3. Round trip info: Create a view that, for each different round trip scheduled for each different date, provides the round trip ID, flight date, destination city code (all round trips originate in Bangor), aircraft ID, pilot ID, plane capacity, total number of reservations (outbound and return trip combined), commercial round trip ticket price, and total annualized profit/loss (assume the week of data given is indicative of all 52 weeks of the year). Sort your final output in ascending order by route # (major field), and flight date (minor field). Hint: each round trip consists of two scheduled flights (the outbound leg and the return leg). You already calculated the profitability of each scheduled flight, so use the Sched_Flights_Profit view (PV#3), summing both the passenger count and the profit, multiplying the summed profit by 52 to annualize it.

RT	FLTDATE	CITY	PLANEID	PILOT	CAP	PASS	RTFARE	PROFIT
1	13-NOV-06	NYL	347665	P9996	10	20	1711	667290
1	17-NOV-06	NYL	347665	P6987	10	12	1711	311402

4. Plane stats: Create a view that, for each aircraft, lists the serial number, type of aircraft, seating capacity, aircraft's fair market value (FMV), number of round trips flown (annualized), total flight hours flown (annualized), and the total annualized profit generated by that aircraft. Hint: join the Aircraft table (for the typecodes and current FMV) with the views you created for flight times and profits (PV#1 and PV#3). To annualize the profits and airtimes, multiply them by 52. However, since there are 2 scheduled flights in a round trip, use a COUNT(*) * 26 to annualize the number of round trips.

PLANE	TYPECODE	CAP	FMV	YRLYTRIPS	YRLYAIRTIME	YRLYPROFIT
101010	GIV	7	975000	260	2470	-73515.52
123456	CMUS	5	450000	52	182	58500

5. Potential London Flight: After signing a multi man-year consulting contract with Lloyd's of London, management wants to explore scheduling TFI flights to London. To help determine the feasibility for such flights, create a view that, for a given date

and a given destination, lists the date, the destination city (LON), distance to Bangor, round trip airtime (assuming 400 mph air speed), serial numbers of all aircraft that could be scheduled for a flight to this destination (i.e., aircraft that have the necessary flight range and are not currently scheduled for those dates), the aircraft's flight range and passenger capacity, and the IDs of the pilots who could be scheduled to fly those round trips (i.e., pilots that have the certification on one or more of the potential aircraft, are not currently scheduled for flights on that date, and whose maximum weekly airtime (30 hours) would not be exceeded by flying this additional round trip), and the pilot's currently scheduled weekly airtime (WAT). Then run this view for the dates that TFI consultants are flying to London. Sort your results in ascending order by aircraft ID and pilot ID. Hint: Create a base table, named PotlFlt, with a single row and 2 attributes: FltDate (DATE format) and Destination city (CHAR(3)), and insert the appropriate FltDate that consultants are currently flying commercial airlines to London ('13-NOV-06') and the destination city ('LON'). Then join the PotlFlt table with your Idle_Planes view (PV#6 – joined on FltDate) and Idle_Pilots (PV#7 – joined on FltDate) and the Certifications table (joined with Idle_Planes on Typecode and Idle_Pilots on PilotID) and the Cities table to get the Dist_to_BGR (joined with PotlFlt on City). Include other conditions as appropriate; i.e., flight range > distance to Bangor, and pilots's weekly airtime + Dist_to_BGR/400*2 less than 30 hours. **Note that you can use this view to explore any new flight by simply changing (with the UPDATE command) the FltDate and City in the PotlFlt table and re-running the view.**

FLTDATE	DEST	DIST	AIRTIME	AIRCRAFT	RANGE	CAP	PILOT	WAT
13-NOV-06	LON	3760	18.8	765765	4000	7	P5432	10
13-NOV-06	LON	3760	18.8	765765	4000	7	P6987	5

6. Overbooked flights: Contrary to commercial airlines, TFI does not allow overbooking of flights (i.e., allowing the number of reservation to exceed the plane's seating capacity). Unfortunately, this sometimes happens in TFI's current reservation system. Thus, management would like a short report that, for any overbooked scheduled flight(s), lists the flight #, flight date, origin and destination, scheduled plane type, scheduled plane's capacity, and the number of passengers currently booked on that flight on that date. Sort your results in ascending order by flight date. Hint: join your Sched_Flights_Profit view (PV#3) with the Flights, Aircraft and Plane_Type tables, and include a HAVING clause to find the 'overbooked' flights.

FLT#	FLTDATE	ORIGIN	DEST	TYPECODE	CAP	PASSNGRS
150	13-NOV-06	BGR	DCA	CMUS	5	6

7. Under booked flights: TFI loses money if consultants fly commercial airlines when there are seats available on TFI flights flying to the same city on the same date. Thus, management would like a listing of the names of all passengers who are flying to a city on the same date that TFI is flying there, along with TFI's corresponding flight number, the flight date, origin city code, destination city code, commercial ticket price, the scheduled aircraft's capacity, and the number of available seats on the TFI flight. Sort your results in ascending order by flight date and TFI flight #. Hint: Join your Sched_Flights_Profit view (PV#3) with the Flights table (joined on Flt#) and the Fly_Comml table (joined on flight date, origin and destination.)

PASSNGR	FLT	FLTDATE	ORI	DEST	FARE	CAP	AVAIL
JONES	138	13-NOV-06	BGR	NYL	1711	10	0
MEADOWS	150	13-NOV-06	BGR	DCA	1500	5	-1

8. Back-up aircraft: In order to aid in the rescheduling of aircraft due to equipment failures or schedule changes, management would like you to generate a view that, for each scheduled flight, lists the flight #, flight date, the distance to Bangor, current number of reservations, and the IDs, passenger capacities and flight ranges of all aircraft that could be used as a back-up; i.e., aircraft that are sufficiently large to carry the passengers scheduled for that flight, have the required flight range for the trip, and are not currently scheduled for a different flight on that date. Sort your output in flt#, fltdate order. Hints: join the Sched_Flights_Profit view (PV#3) with Idle_Aircraft view (PV#7 – joined on flight date) and the Cities table (joined on city). Include other conditions as appropriate. (See below for the first few rows of the output)

FLT	FLTDATE	DIST	PASS	PLANE	CAP	RANGE
130	13-NOV-06	1950	3	987987	7	2700
130	13-NOV-06	1950	3	765765	7	4000

APPENDIX 3.
TERRA FIRMA AIRLINES - QUERIES B

After creating the database and loading the test data, TFI requests that you use the data to generate the appropriate responses to the Executive Committee's fifteen questions (see below). Specifically, for each query, list (and clearly label with the appropriate query number) the **final query output only**. Then, in an appendix, list (and clearly label with the appropriate query number) the views and queries that were used to generate the final output. TFI accountants have laboriously calculated and provided some numbers that you can use to check your output – these check-sums are provided after most queries and usually consist of the first few rows of the output you should expect to get; for those queries without check-sums, you'll need to do some checking on your own to validate your results. Unless otherwise specified in the query, you should round all dollar amounts to the nearest penny, and all percentages to the nearest tenth of a percent. Also, answer any questions below by filling in the blanks provided, and handing in these pages as the cover sheets of your output packet. That is, each team should hand in one answer sheet, one set of final query outputs, and one appendix.

1. Overbooked Passengers: Create a view that, for each passenger with reservations for 2 or more round trips on any given date, lists the passenger name, round trip IDs, flight numbers, flight dates, origin cities, ETDs, destination cities, and ETAs. Sort your results in ascending order by passenger name, round trip ID and ETD. Hint: Join the Reservations table with your Overbooked_Passengers view (PV#8 – on FltDate and Passngr) and the Flights table (on Flt#).

PASSNGR	RTID	FLT#	FLTDATE	ORIGIN	ETD	DEST	ETA
CAROLE	1	138	13-NOV-06	BGR	630	NYL	745
CAROLE	1	211	13-NOV-06	NYL	1945	BGR	2100
CAROLE	8	170	13-NOV-06	BGR	700	DFW	1145

2. Pilot certifications: Create a view that, for each scheduled flight for which the pilot is NOT certified to fly the scheduled aircraft, lists the flight number, flight date, aircraft serial number, aircraft typecode, pilot ID, and all aircraft types that the pilot is certified to fly. Sort your results in ascending order by flight number and flight date. Hint: Join the Certifications table with your Mismatches view (PV#9 – on PilotID).

FLT#	FLTDATE	AIRCRAFTID	TYPECODE	PILOTID	CERT
170	15-NOV-06	101010	GIV	P6546	CMUS
170	15-NOV-06	101010	GIV	P6546	LJ40

3. Plane maintenance: Create a view that, for each aircraft, lists the aircraft ID, last service date, flight hours between service, hours flown since last service date (assume Nov. 14-20 is indicative of all weeks), average weekly flight hours, and number of hours until next maintenance, and the approximate number of weeks until the next maintenance should be scheduled. Round all calculations to the nearest tenth. Sort your results in ascending order by number of weeks until next scheduled maintenance. Hint: to calculate the hours flown since the last service date (HRS), calculate the number of weeks since the last service date and multiply by the weekly airtime in your WklyAirtime view (PV#10), using CURRENT_DATE (a reserved SQL variable that always contains today's date) -- this is called 'date arithmetic' in SQL; e.g.,

```
SELECT AIRCRAFTID,(CURRENT_DATE-LAST_SVC_DATE)/7*WKLYHRS HRS
FROM WKLYAIRTIME;
```

The Hours_until_next_maintenance (HRSNEXT) is simply time_between_service (TBS) minus the Hours_since_last_service (HRS). Finally, the Weeks_til_next_maintenance (WKSNEXT) is Hours_until_next_maintenance (HRSNEXT) divided by the weekly airtime (WKLYHRS). You will get different numbers than those shown below because our CURRENT_DATES are not the same. Further, if your HRSNEXT is less than zero, it simply indicates that it is past the plane's normal maintenance date.

PLANE	LSD	TBS	HRS	WKLYHRS	HRSNEXT	WKSNEXT
101010	11-OCT-06	200	355.5	47.5	-155.5	-3.3
235440	11-NOV-06	250	42.8	14	207.2	14.8

4. Overworked Pilots: A pilot is 'overworked' if s/he is scheduled for more than 30 hours of airtime in a week. Create a view that, for each pilot, provides the pilot's name, ID, the number of hours s/he is scheduled to fly during the representative week, and the percent of the 30 hours work week s/he is scheduled to fly (rounded to the nearest whole number). Sort your results in order of scheduled flying time, lowest first. Hint: Join the Sched_Flights table with the Flights_Plus view (PV#1 – on Flt#) and the Pilots table (on PilotID).

NAME	PILOTID	TOTAIRTIME	PERCENT
KENNEDY	P9078	3.5	12
EDLEMAN	P6987	5	17

5. Potential LA flight: Management wants to explore adding a TTI flight for the outbound and return trip to LAX. Create a view that lists the flight date, destination city, the distance from Bangor, the airtime to LAX (assuming an average speed of 400 mph), the serial numbers of the aircraft that could be scheduled for a LAX flight on 14-NOV-06 (i.e., that have the necessary flight range and are not currently scheduled for that date), the aircraft's range and capacity, the IDs and current weekly airtime of the pilots who could be scheduled (i.e., pilots that have the certification on one or more of the potential aircraft and are not currently scheduled for that date and have sufficient slack in their weekly air time to avoid the 30 hour maximum after adding the LAX trip). Sort your results in ascending order by aircraft ID and pilot ID. Hint: this is similar to #5 in Queries A – see the bolded comment at the bottom of QA#5.

FLTDATE	DEST	DIST	AIRTIME	AIRCRAFT	RANGE	CAP	PILOT	WAT
14-NOV-06	LAX	3170	15.85	101010	4000	7	P5432	10
14-NOV-06	LAX	3170	15.85	101010	4000	7	P6987	5

6. Potential layovers: Create a view that, for each city TFI flies to (other than Bangor), lists the round trip ID, destination city code, the number of TFI reservations made each week to that city, and the average number of days that passengers stay there before flying back. Sort your results in order by round trip ID. Hint: Join your Inbound_Passengers and Outbound_Passengers views (PV#11 & PV#12 – on Passngr and RTID) and use a correlated subquery to find the minimum inbound flight date, correlated on the outbound flight date, passenger name, and roundtrip ID.

RTID	DEST	NUMPASS	AVGSTAY
1	NYL	16	0
2	DCA	11	1.5

7. Back-up pilots: Create a view that, for each scheduled flight, provides the flight #, flight date, currently assigned pilot ID, assigned aircraft ID, type of aircraft, and the IDs of all pilots who could serve as an alternate pilot; i.e., pilots that are certified to fly the type of aircraft scheduled, whose total airtime/week would not exceed 30 hours if s/he flew this additional flight, **and** who do not have a scheduling conflict (scheduled for a different flight on that date). Also include in your view the current weekly airtime for each alternate pilot and the airtime of the scheduled roundtrip. Sort your results by FltDate, Flt#, and RoundtripID. Hint: Join your Idle_Pilots view (PV#6) and the Sched_Flights table (on Fltdate) and the Aircraft table (on Flt#) to list the required info. One way to incorporate the 3 conditions for alternate pilots is to use 3 subqueries, one for each of the conditions. You can assume that a roundtrip airtime is twice its corresponding 1-way airtime. The first few rows of output are:

FLT	FLTDATE	PILOT	PLANE	TYPE	ALT	TAT	AIR
130	13-NOV-06	P1234	101010	GIV	P6987	5	9.5
130	13-NOV-06	P1234	101010	GIV	P5432	10	9.5

8. Demand driven plane changes: Create a view that includes both the TFI reservations and commercial fliers scheduled for a trip on the same date to the same destination as one of the TFI scheduled flights. In this view, provide the flight #, flight date, scheduled aircraft's ID, type and capacity, and the total number of potential passengers (from both the Reservations and Fly_Comml tables) that have, or could, make a reservation on that scheduled flight. Then list the information for those flights whose potential number of passengers exceeds the aircraft's capacity. Sort your output in descending order by the number of excess passengers (# passengers less the aircraft's capacity). Hint: Join the Sched_Flights table with the Flights table (on Flt#) and your All_Passngrs view (PV#13 – on FltDate, Origin and Destination) and the Aircraft table (on AircraftID) and the Plane_Type table (on TypeCode).

FLT#	FLTDATE	PLANE	TYPE	CAP	PASSNGRS
175	17-NOV-06	101010	GIV	7	11
135	17-NOV-06	101010	GIV	7	11

9. Create a view named `Income Stmt` that includes the following income statement calculations:

- a. total revenues from reservations (`REVR`). Hint: use the `TFI_INC` view created in `PV#14`.
- b. total revenues from the `Fly_Comml` table (`REVC`) – the total revenue generated by TFI consultants flying on commercial airlines (equals $0.5 * \text{round trip commercial airfares}$ summed over all rows in the `Fly_Comml` table). Hint: join the `Fly_Comml` and `Cities` tables using:

`WHERE(Origin = City OR Destination = City)`
`AND RTComm_Airfare > 0`

- c. total variable costs of TFI scheduled flights (`EXPR`). Hint: use your `TFI_INC` view created in `PV#14`.
- d. total cost of all commercial tickets TFI bought for their consultants (`EXPC`) (this is the same as b above; i.e., since TFI Air makes no profit from consultants who fly commercial airlines, the expense equals the revenue).
- e. aircraft depreciation expense (`DEPR`) – the depreciation of all aircraft based on a 5 year straight line depreciation schedule using the fair market value (`FMV`) in the `Aircraft` table (Hint: divide the `FMV` by 5 and sum it for all aircraft in the `Aircraft` table).
- f. total pilot expense (`PILOT`). Hint: use your `TFI_INC` view created in `PV#14`.
- g. earnings before interest and taxes (`EBIT`) – add a) and b); then subtract c), d), e), and f).

<code>REVT</code>	<code>REVC</code>	<code>TOTREV</code>	<code>TFIEXP</code>	<code>EXPC</code>	<code>DEPR</code>	<code>PILOT</code>	<code>TOTEXP</code>	<code>EBIT</code>
4173312	6012552	10185864	2558465	6012552	879000	716800	10166817	19047



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

Copyright ©2008 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, Journal of Information Systems Education, editor@jise.org.

ISSN 1055-3096