```
--create dimensional tables

--create fact table



CREATE TABLE PILOT_DIM

(
PILOTID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
EMP_NUM INT NOT NULL,
PIL_LICENSE NVARCHAR(25),
PIL_RATINGS NVARCHAR(25),
PIL_MED_TYPE NVARCHAR(1),
PIL_MED_DATE DATETIME,
PIL_PT135_DATE DATETIME,
EMP_TITLE NVARCHAR(4),
EMP_LNAME NVARCHAR(15),
EMP_FNAME NVARCHAR(15),
EMP_INITIAL NVARCHAR(1),
EMP_DOB DATETIME,
EMP_HIRE_DATE DATETIME
)


CREATE TABLE MODEL_DIM

(
ModelID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
MOD_CODE NVARCHAR(10) NOT NULL,
MOD_MANUFACTURER NVARCHAR(15),
MOD_NAME NVARCHAR(20),
MOD_SEATS FLOAT,
MOD_CHG_MILE REAL,
MOD_CRUISE FLOAT,
MOD_FUEL FLOAT
)


CREATE TABLE TIME_DIM

(
TIMEID              INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
DATEVALUE           DATETIME,
YEAR_NUM            INT,
MONTH_NUM           INT,
DAY_NUM                    INT,
)
```

```
CREATE TABLE FACT_TABLE

(
ModelID INT NOT NULL,
TimeID INT NOT NULL,
PilotID INT NOT NULL,
TOT_HOURS_FLOWN FLOAT,
TOT_FUEL_USED FLOAT,
TOT_REVENUE REAL
)




ALTER TABLE FACT_TABLE
ADD CONSTRAINT FK_TIME FOREIGN KEY (TimeID) REFERENCES TIME_DIM
ALTER TABLE FACT_TABLE
ADD CONSTRAINT FK_MODEL FOREIGN KEY (ModelID) REFERENCES MODEL_DIM
ALTER TABLE FACT_TABLE
ADD CONSTRAINT FK_PILOT FOREIGN KEY (PilotID) REFERENCES PILOT_DIM


ALTER TABLE FACT_TABLE
ADD PRIMARY KEY (ModelID, TimeID, PilotID)


--execute stored procedure
EXEC A11 --EXECUTES CORRECTLY


--check tables
select *
from time_dim

select *
from pilot_dim


select *
from model_dim


select *
from fact_table


--- for questions
SELECT T.YEAR_NUM, T.MONTH_NUM, T.DAY_NUM, SUM(F.TOT_HOURS_FLOWN)
FROM FACT_TABLE F INNER JOIN .TIME_DIM T ON F.TIMEID = T.TIMEID
GROUP BY T.YEAR_NUM, T.MONTH_NUM, T.DAY_NUM
ORDER BY T.YEAR_NUM, T.MONTH_NUM, T.DAY_NUM
```

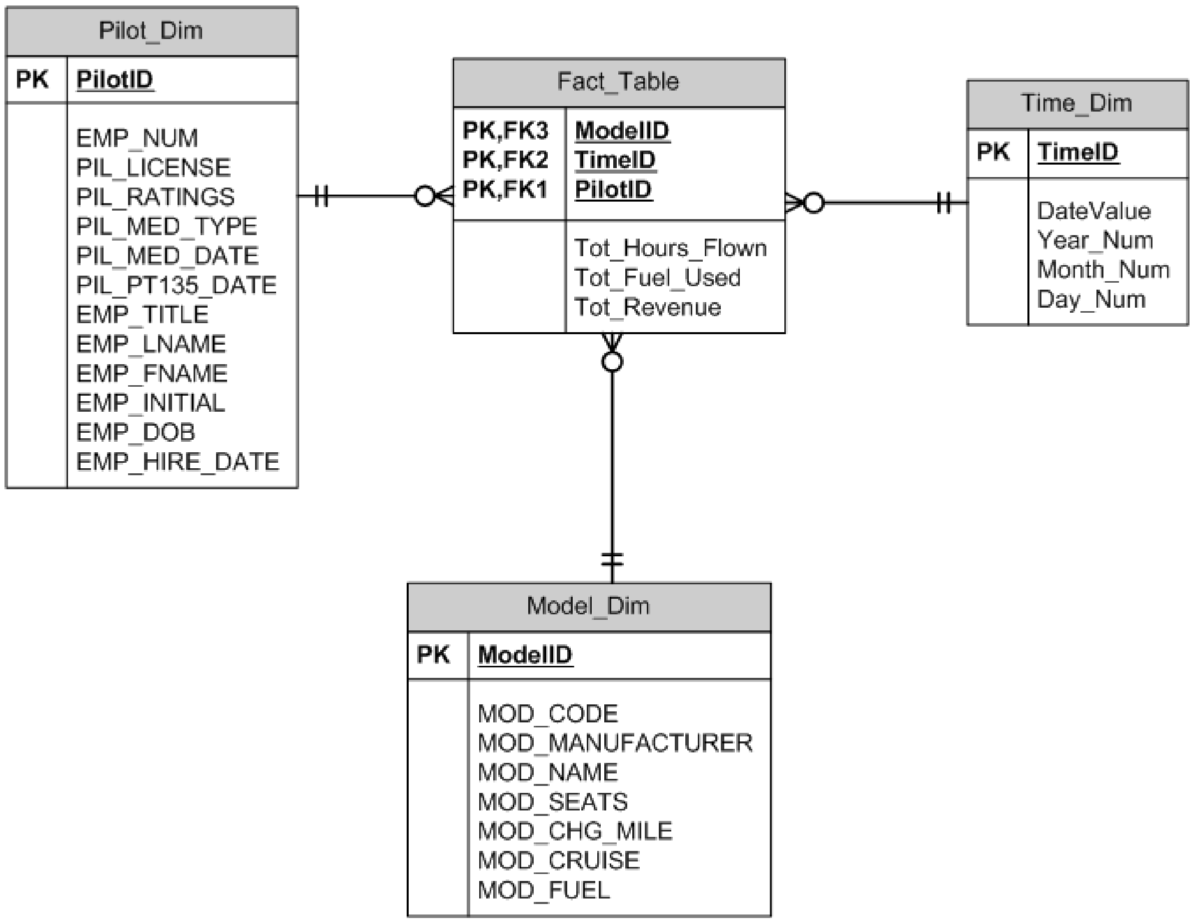
```sql
ALTER VIEW YEARVIEW AS
SELECT T.YEAR_NUM,  MAX(F.TOT_HOURS_FLOWN) AS 'Y TOT HOURS FLOWN', T.TIMEID
FROM FACT_TABLE F INNER JOIN TIME_DIM T ON F.TIMEID = T.TIMEID
GROUP BY T.YEAR_NUM, T.TIMEID


ALTER VIEW MONTHVIEW AS
SELECT T.MONTH_NUM,  MAX(F.TOT_HOURS_FLOWN) AS 'M TOT HOURS FLOWN', T.TIMEID
FROM FACT_TABLE F INNER JOIN TIME_DIM T ON F.TIMEID = T.TIMEID
GROUP BY T.MONTH_NUM, T.TIMEID


---------- WHAT IS THE MAX SUMMED TOTAL HOURS FLOWN PER YEAR, MONTH, AND DAY

CREATE VIEW YVIEW AS
SELECT T.YEAR_NUM, SUM(F.TOT_HOURS_FLOWN) AS 'YEAR SUM'
FROM TIME_DIM T INNER JOIN FACT_TABLE F ON T.TIMEID = F.TIMEID
GROUP BY T.YEAR_NUM

CREATE VIEW MVIEW AS
SELECT T.MONTH_NUM, SUM(F.TOT_HOURS_FLOWN) AS 'MONTH SUM'
FROM TIME_DIM T INNER JOIN FACT_TABLE F ON T.TIMEID = F.TIMEID
GROUP BY T.MONTH_NUM


CREATE VIEW DVIEW AS
SELECT T.DAY_NUM,SUM(F.TOT_HOURS_FLOWN) AS 'DAY SUM'
FROM TIME_DIM T INNER JOIN FACT_TABLE F ON T.TIMEID = F.TIMEID
GROUP BY T.DAY_NUM


CREATE VIEW DMAX AS
SELECT DISTINCT DAY_NUM, MAX([DAY SUM]) AS 'MAX_DAY'
FROM DVIEW
GROUP BY DAY_NUM


CREATE VIEW V1 AS
SELECT  DAY_NUM, MAX([DAY SUM]) AS 'MAX_DAY'
FROM DVIEW
WHERE [DAY SUM] = (SELECT MAX([DAY SUM]) AS 'MDAY' FROM DVIEW)
GROUP BY DAY_NUM


CREATE VIEW V2 AS
SELECT MONTH_NUM, MAX([MONTH SUM]) AS 'MAX_MONTH'
FROM MVIEW
WHERE [MONTH SUM] = (SELECT MAX([MONTH SUM]) AS 'MMONTH' FROM MVIEW)
GROUP BY MONTH_NUM
```

```
CREATE VIEW V3 AS
SELECT YEAR_NUM, MAX([YEAR SUM]) AS 'MAX_YEAR'
FROM YVIEW
WHERE [YEAR SUM] = (SELECT MAX([YEAR SUM]) AS 'MYEAR' FROM YVIEW)
GROUP BY YEAR_NUM


SELECT YEAR_NUM, MAX_YEAR, MONTH_NUM, MAX_MONTH, DAY_NUM, MAX_DAY
FROM V1 CROSS JOIN V2 CROSS JOIN V3


--what is the average TOTAL HOURS FLOWN AS PER YEAR, MONTH, DAY

SELECT T.YEAR_NUM, AVG(F.TOT_HOURS_FLOWN)
FROM FACT_TABLE F INNER JOIN TIME_DIM T ON F.TIMEID = T.TIMEIID
GROUP BY Y.YEAR_NUM


-----UPDATING QUESTION


-- ================================================
-- Template generated from Template Explorer using:
-- Create Procedure (New Menu).SQL
--
-- Use the Specify Values for Template Parameters
-- command (Ctrl-Shift-M) to fill in the parameter
-- values below.
--
-- This block of comments will not be included in
-- the definition of the procedure.
-- ================================================
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:		<Author,,Name>
-- Create date: <Create Date,,>
-- Description:	<Description,,>
-- =============================================
CREATE PROCEDURE A11
	-- Add the parameters for the stored procedure here
--	<@Param1, sysname, @p1> <Datatype_For_Param1, , int> = <Default_Value_For_Param1, , 0>,
--	<@Param2, sysname, @p2> <Datatype_For_Param2, , int> = <Default_Value_For_Param2, , 0>
AS
BEGIN
	-- SET NOCOUNT ON added to prevent extra result sets from
	-- interfering with SELECT statements.
	SET NOCOUNT ON;

--turn off contraints
ALTER TABLE FACT_TABLE DROP CONSTRAINT FK_TIME
ALTER TABLE FACT_TABLE DROP CONSTRAINT FK_MODEL
ALTER TABLE FACT_TABLE DROP CONSTRAINT FK_PILOT
ALTER TABLE PILOT_DIM NOCHECK CONSTRAINT ALL
ALTER TABLE MODEL_DIM NOCHECK CONSTRAINT ALL
ALTER TABLE TIME_DIM NOCHECK CONSTRAINT ALL

--remove all rows
TRUNCATE TABLE FACT_TABLE
TRUNCATE TABLE PILOT_DIM
TRUNCATE TABLE MODEL_DIM
TRUNCATE TABLE TIME_DIM


--turn contraints back on
ALTER TABLE PILOT_DIM CHECK CONSTRAINT ALL
ALTER TABLE MODEL_DIM CHECK CONSTRAINT ALL
ALTER TABLE FACT_TABLE CHECK CONSTRAINT ALL
ALTER TABLE TIME_DIM CHECK CONSTRAINT ALL
ALTER TABLE FACT_TABLE
ADD CONSTRAINT FK_MODEL FOREIGN KEY (ModelID) REFERENCES MODEL_DIM
ALTER TABLE FACT_TABLE
ADD CONSTRAINT FK_TIME FOREIGN KEY (TimeID) REFERENCES TIME_DIM
ALTER TABLE FACT_TABLE
ADD CONSTRAINT FK_PILOT FOREIGN KEY (PilotID) REFERENCES PILOT_DIM


    -- Insert statements for procedure here

--place information into dimensional table
INSERT INTO PILOT_DIM
SELECT E.EMP_NUM, P.PIL_LICENSE, P.PIL_RATINGS, P.PIL_MED_TYPE, P.PIL_MED_DATE, P.PIL_PT135_DATE,
E.EMP_TITLE, E.EMP_LNAME, E.EMP_FNAME, E.EMP_INITIAL, E.EMP_DOB, E.EMP_HIRE_DATE
FROM PILOT P INNER JOIN EMPLOYEE E ON P.EMP_NUM = E.EMP_NUM

INSERT INTO MODEL_DIM
SELECT MOD_CODE, MOD_MANUFACTURER, MOD_NAME, MOD_SEATS, MOD_CHG_MILE, MOD_CRUISE, MOD_FUEL
FROM MODEL

INSERT INTO TIME_DIM (DATEVALUE, YEAR_NUM, MONTH_NUM, DAY_NUM)
SELECT DISTINCT CHAR_DATE, YEAR(CHAR_DATE), MONTH(CHAR_DATE), DAY(CHAR_DATE)
FROM CHARTER

--create the transitional table
CREATE TABLE STAGE
(
MODELID                 INT,
MOD_CODE        NVARCHAR(10),
TIMEID          INT,
CHAR_DATE       DATETIME,
PILOTID         INT,
EMP_NUM                 INT,
CHAR_HOURS_FLOWN FLOAT,
CHAR_FUEL_GALLONS FLOAT,
CHAR_DISTANCE INT,
MOD_CHG_MILE          REAL,
CHAR_PILOT INT,
AC_NUMBER NVARCHAR(5)
)

--insert transactional data to Stage
INSERT INTO STAGE (CHAR_DATE, MOD_CODE, EMP_NUM, CHAR_PILOT, AC_NUMBER, CHAR_HOURS_FLOWN,
CHAR_FUEL_GALLONS, CHAR_DISTANCE, MOD_CHG_MILE)
SELECT C.CHAR_DATE, M.MOD_CODE, P.EMP_NUM, C.CHAR_PILOT, C.AC_NUMBER, C.CHAR_HOURS_FLOWN,
C.CHAR_FUEL_GALLONS, C.CHAR_DISTANCE, M.MOD_CHG_MILE
FROM PILOT P INNER JOIN CHARTER C ON C.CHAR_PILOT = P.EMP_NUM INNER JOIN AIRCRAFT A ON C.AC_NUMBER =
A.AC_NUMBER INNER JOIN MODEL M ON A.MOD_CODE = M.MOD_CODE


--update the stage table to sync with Identities of the dimensional tables
UPDATE STAGE
SET MODELID = M.MODELID
FROM STAGE S INNER JOIN MODEL_DIM M ON S.MOD_CODE = M.MOD_CODE


UPDATE STAGE
SET PILOTID = P.PILOTID
FROM STAGE S INNER JOIN PILOT_DIM P ON S.EMP_NUM = P.EMP_NUM


UPDATE STAGE
SET TIMEID = T.TIMEID
FROM STAGE S INNER JOIN TIME_DIM T ON S.CHAR_DATE = T.DATEVALUE

--place inforation from stage table into fact table including calculations but not aggregates
INSERT INTO FACT_TABLE
SELECT MODELID, TIMEID, PILOTID, CHAR_HOURS_FLOWN, CHAR_FUEL_GALLONS, (CHAR_DISTANCE *
MOD_CHG_MILE) AS 'TOT_REVENUE'
FROM STAGE

--remove the stage table
DROP TABLE STAGE


END
GO


QUESTIONS FROM BOOK

--A) STAR SCHEMA IS INCLUDED.

--B) define the dimensions and attributes for the charter data
CREATE TABLE PILOT_DIM
(
PILOTID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
EMP_NUM INT NOT NULL,
PIL_LICENSE NVARCHAR(25),
PIL_RATINGS NVARCHAR(25),
PIL_MED_TYPE NVARCHAR(1),
```

```sql
PIL_MED_DATE DATETIME,
PIL_PT135_DATE DATETIME,
EMP_TITLE NVARCHAR(4),
EMP_LNAME NVARCHAR(15),
EMP_FNAME NVARCHAR(15),
EMP_INITIAL NVARCHAR(1),
EMP_DOB DATETIME,
EMP_HIRE_DATE DATETIME
)

CREATE TABLE MODEL_DIM
(
ModelID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
MOD_CODE NVARCHAR(10) NOT NULL,
MOD_MANUFACTURER NVARCHAR(15),
MOD_NAME NVARCHAR(20),
MOD_SEATS FLOAT,
MOD_CHG_MILE REAL,
MOD_CRUISE FLOAT,
MOD_FUEL FLOAT
)

CREATE TABLE TIME_DIM
(
TIMEID                  INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
DATEVALUE         DATETIME,
YEAR_NUM          INT,
MONTH_NUM         INT,
DAY_OF_MONTH      INT,
)

CREATE TABLE FACT_TABLE
(
ModelID INT NOT NULL,
TimeID INT NOT NULL,
PilotID INT NOT NULL,
TOT_HOURS_FLOWN FLOAT,
TOT_FUEL_USED FLOAT,
TOT_REVENUE REAL
)

--C define the necessary attribute hierarchies
YEAR_NUM -> MONTH_NUM -> DAY_NUM

--D  implement the data warehouse design, using the design components you developed in
prob 4a-4c
see data warehouse

--E  3 questions pertaining to each dimensional table

--What is each pilot's total revenue and total hours flown and total gallons used?
SELECT PILOTID, ROUND(SUM(TOT_REVENUE),2) AS 'TOTAL REVENUE', SUM(TOT_HOURS_FLOWN) AS
'TOTAL HOURS FLOWN', ROUND(SUM(TOT_FUEL_USED), 2) AS 'TOTAL FUEL USED'
FROM FACT_TABLE
GROUP BY PILOTID
ORDER BY PILOTID

--What are the total hours flown and the total revenue for each day?
SELECT TIMEID, SUM(TOT_HOURS_FLOWN) AS 'TOTAL HOURS FLOWN', ROUND(SUM(TOT_REVENUE),2) AS
'TOTAL REVENUE'
FROM FACT_TABLE
GROUP BY TIMEID
ORDER BY TIMEID
```

```
--What is the total amount of fuel each plane uses and what are the total hours flown and
total revenue?
SELECT MODELID, ROUND(SUM(TOT_HOURS_FLOWN),2) AS 'TOTAL HOURS FLOWN',
ROUND(SUM(TOT_FUEL_USED), 2) AS 'TOTAL FUEL USED', ROUND(SUM(TOT_REVENUE),2) AS 'TOTAL
REVENUE'
FROM FACT_TABLE
GROUP BY MODELID
ORDER BY MODELID



--question that involves all tables

--What are the total hours flown, total fuel used, total revenue by each pilot that flies
a Beechcraft in 2004

SELECT F.PILOTID, F.MODELID, SUM(TOT_HOURS_FLOWN) AS 'TOTAL HOURS FLOWN',
SUM(TOT_FUEL_USED) AS 'TOTAL FUEL USED', ROUND(SUM(TOT_REVENUE),2) AS 'TOTAL REVENUE',
T.YEAR_NUM, M.MOD_MANUFACTURER
FROM FACT_TABLE F INNER JOIN TIME_DIM T ON F.TIMEID = T.TIMEID INNER JOIN PILOT_DIM P ON
F.PILOTID = P.PILOTID INNER JOIN MODEL_DIM M ON F.MODELID = M.MODELID
WHERE M.MOD_MANUFACTURER = 'Beechcraft' AND T.YEAR_NUM = 2004
GROUP BY F.PILOTID, F.MODELID, T.YEAR_NUM, M.MOD_MANUFACTURER


--question that involves aggregation and with a dimensional hierarchy

---------- WHAT IS THE MAX SUMMED TOTAL HOURS FLOWN, FUEL USED, AND REVENUE PER YEAR,
MONTH, AND DAY - assuming that the maximum amount of flight times uses the same gas and
there are not sales

CREATE VIEW YALLVIEW AS
SELECT T.YEAR_NUM, SUM(F.TOT_HOURS_FLOWN) AS 'FLOWN YEAR SUM', SUM(F.TOT_FUEL_USED) AS
'FUEL YEAR SUM', SUM(F.TOT_REVENUE) AS 'REV YEAR SUM'
FROM TIME_DIM T INNER JOIN FACT_TABLE F ON T.TIMEID = F.TIMEID
GROUP BY T.YEAR_NUM

CREATE VIEW VALLY AS
SELECT YEAR_NUM, MAX([FLOWN YEAR SUM]) AS 'FLOWN MAX YEAR', MAX([FUEL YEAR SUM]) AS 'FUEL
MAX YEAR', SUM([REV YEAR SUM]) AS 'REV MAX YEAR'
FROM YALLVIEW
WHERE [FLOWN YEAR SUM] = (SELECT MAX([FLOWN YEAR SUM]) AS 'MYEAR' FROM YALLVIEW)
GROUP BY YEAR_NUM

CREATE VIEW MALLVIEW AS
SELECT T.MONTH_NUM, SUM(F.TOT_HOURS_FLOWN) AS 'FLOWN MONTH SUM', SUM(F.TOT_FUEL_USED) AS
'FUEL MONTH SUM', SUM(F.TOT_REVENUE) AS 'REV MONTH SUM'
FROM TIME_DIM T INNER JOIN FACT_TABLE F ON T.TIMEID = F.TIMEID
GROUP BY T.MONTH_NUM

CREATE VIEW VALLM AS
SELECT MONTH_NUM, MAX([FLOWN MONTH SUM]) AS 'FLOWN MAX MONTH', MAX([FUEL MONTH SUM]) AS
'FUEL MAX MONTH', MAX([REV MONTH SUM]) AS 'REV MAX MONTH'
FROM MALLVIEW
WHERE [FLOWN MONTH SUM] = (SELECT MAX([FLOWN MONTH SUM]) AS 'MMONTH' FROM MALLVIEW)
GROUP BY MONTH_NUM

CREATE VIEW DALLVIEW AS
SELECT T.DAY_NUM,SUM(F.TOT_HOURS_FLOWN) AS 'FLOWN DAY SUM', SUM(F.TOT_FUEL_USED) AS 'FUEL
DAY SUM', SUM(F.TOT_REVENUE) AS 'REV DAY SUM'
FROM TIME_DIM T INNER JOIN FACT_TABLE F ON T.TIMEID = F.TIMEID
GROUP BY T.DAY_NUM
```

```
CREATE VIEW DAYMAX AS
SELECT DISTINCT DAY_NUM, MAX([FLOWN DAY SUM]) AS 'FLOWN MAX DAY', MAX([FUEL DAY SUM]) AS
'FUEL MAX DAY', MAX([REV DAY SUM]) AS 'REV MAX DAY'
FROM DALLVIEW
WHERE [FLOWN DAY SUM] = (SELECT MAX([FLOWN DAY SUM]) AS 'DMONTH' FROM DALLVIEW)
GROUP BY DAY_NUM

SELECT YEAR_NUM, [FLOWN MAX YEAR], [FUEL MAX YEAR], [REV MAX YEAR], MONTH_NUM, [FLOWN MAX
MONTH], [FUEL MAX MONTH], [REV MAX MONTH], DAY_NUM, [FLOWN MAX DAY], [FUEL MAX DAY], [REV
MAX DAY]
FROM DAYMAX CROSS JOIN VALLY CROSS JOIN VALLM
```


Pilot_Dim
PK
PilotID
EMP_NUM
PIL_LICENSE
PIL_RATINGS
PIL_MED_TYPE
PIL_MED_DATE
PIL_PT135_DATE
EMP_TITLE
EMP_LNAME
EMP_FNAME
EMP_INITIAL
EMP_DOB
EMP_HIRE_DATE
Fact_Table
PK,FK3
PK,FK2
PK,FK1
ModelID
TimeID
PilotID
Tot_Hours_Flown
Tot_Fuel_Used
Tot_Revenue
Time_Dim
PK
TimeID
DateValue
Year_Num
Month_Num
Day_Num
Model_Dim
PK
ModelID
MOD_CODE
MOD_MANUFACTURER
MOD_NAME
MOD_SEATS
MOD_CHG_MILE
MOD_CRUISE
MOD_FUEL