

# ICS 311 - Database Management Systems

## Assignment 5

Due: See Syllabus

### Question 1: Normalization

Consider a relation R with six attributes ABCDEF where AB is a composite primary key for R. Assume R has the following functional dependencies (R is one table with the attributes A, B, C, D, E, F):

1.1)

AB  $\rightarrow$  CDEF  
B  $\rightarrow$  CDE  
D  $\rightarrow$  E

a) In what normal form is relation R? Explain why?

The relation is in 1<sup>st</sup> NF because the relation contains partial functional dependency (B  $\rightarrow$  CDE).

b) Is it possible to decompose R into a number of relations in order to achieve a higher normal form? Explain how (this means using English sentences and a diagram showing the final relation(s)).

It's possible for relation R to achieve higher form, first to 2NF and then to 3NF.

To first achieve 2NF, break into 2 tables using the given functional dependencies. Table1 would have AB  $\rightarrow$  F and table2 would have B  $\rightarrow$  CDE

Table1

A	B	F

Table2

B	C	D	E

To achieve 3NF, break dependency.

B  $\rightarrow$  CD and D  $\rightarrow$  E

Table2 into 2 tables because it has a transitive

Table1

A	B	F

Table2

B	C	D

Table3

D	E

1.2)

branchNo	branchAddress	telNo	mgrStaffNo	name
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618	S1500	Tom Daniels
B002	City Center Plaza, Seattle, WA 98122	206-555-6756	S0010	Mary Martinez
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000	S0145	Art Peters
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131	S2250	Sally Stern

Primary key: branchNo

a) Why the above table is not in 3NF?

This table is not 3NF, because we can still achieved atomicity on branchAddress and name. In addition there is transitive functional dependency (mgrStaffNo  $\square$  name)

b) Describe and illustrate the process of normalizing the data shown in this table to third normal form (3NF).

To first move it to 1NF, break branchAddress down into 4 different columns (branchStreet, branchCity, branchState, branchZip), and break name into 2 columns (first\_name, last\_name).

To move it to 2NF and 3NF, split the table to 2 tables with (branchNo  $\square$  branchStreet, branchCity, branchState, branchZip, telNo) and (mgrStaffNo  $\square$  first\_name, last\_name).

branchNo will be a foreign key on the Manager table.

c) Identify the primary and foreign keys in your 3NF relations. Note - There are other correct options for this.

### Branch

<u>branchNo</u>	branchStreet	branchCity	branchState	branch_zip	telNo
B001	8 Jefferson Way	Portland	OR	97201	503-555-3618
B002	City Center Plaza	Seattle	WA	98122	206-555-6756
B003	14 – 8th Avenue	New York	NY	10012	212-371-3000
B004	16 – 14th Avenue	Seattle	WA	98128	206-555-3131

### Manager

<u>mgrStaffNo</u>	first_name	last_name	branchNo
S1500	Tom	Daniels	B001
S0010	Mary	Martinez	B002
S0145	Art	Peters	B003

S2250	Sally	Stern	B004
-------	-------	-------	------

## Question 2: Triggers and Stored Procedures

Table name: CHARTER										Database name: Ch08_AviaCo		
CHAR_TRIP	CHAR_DATE	AC_NUMBER	CHAR_DESTINATION	CHAR_DISTANCE	CHAR_HOURS_FLOWN	CHAR_HOURS_WAIT	CHAR-TOT-CHG	HAR_OIL_QTS	CUS_CODE			
10001	05-Feb-08	2289L	ATL	936	5.1	2.2	354.1	1	10011			
10002	05-Feb-08	2778V	BNA	320	1.6	0	72.6	0	10016			
10003	05-Feb-08	4278Y	GNV	1574	7.8	0	339.8	2	10014			
10004	06-Feb-08	1484P	STL	472	2.9	4.9	97.2	1	10019			
10005	06-Feb-08	2289L	ATL	1023	5.7	3.5	397.7	2	10011			
10006	06-Feb-08	4278Y	STL	472	2.6	5.2	117.1	0	10017			
10007	06-Feb-08	2778V	GNV	1574	7.9	0	348.4	2	10012			
10008	07-Feb-08	1484P	TYS	644	4.1	0	140.6	1	10014			
10009	07-Feb-08	2289L	GNV	1574	6.6	23.4	459.9	0	10017			
10010	07-Feb-08	4278Y	ATL	998	6.2	3.2	279.7	0	10016			
10011	07-Feb-08	1484P	BNA	352	1.9	5.3	66.4	1	10012			
10012	08-Feb-08	2778V	MOB	884	4.8	4.2	215.1	0	10010			
10013	08-Feb-08	4278Y	TYS	644	3.9	4.5	174.3	1	10011			
10014	09-Feb-08	4278Y	ATL	936	6.1	2.1	302.6	0	10017			
10015	09-Feb-08	2289L	GNV	1645	6.7	0	459.5	2	10016			
10016	09-Feb-08	2778V	MOY	312	1.5	0	67.2	0	10011			
10017	10-Feb-08	1484P	STL	508	3.1	0	105.5	0	10014			
10018	10-Feb-08	4278Y	TYS	644	3.8	4.5	167.4	0	10017			

  

Table name: EARNEDRATING		
EMP_NUM	RTG_CODE	EARNRTG_DATE
101	CFI	18-Feb-98
101	CFI	15-Dec-05
101	INSTR	08-Nov-93
101	MEL	23-Jun-94
101	SEL	21-Apr-93
104	INSTR	15-Jul-96
104	MEL	29-Jan-97
104	SEL	12-Mar-95
105	CFI	19-Nov-97
105	INSTR	17-Apr-95
105	MEL	12-Aug-95
105	SEL	23-Sep-94
106	INSTR	20-Dec-95
106	MEL	02-Apr-96
106	SEL	10-Mar-94
109	CFI	05-Nov-98
109	CFI	21-Jun-03
109	INSTR	23-Jul-96
109	MEL	15-Mar-97
109	SEL	05-Feb-96
109	SES	12-May-96

  

Table name: CREW			Table name: CUSTOMER							
CHAR_TRIP	EMP_NUM	CREW_JOB	CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_BALANCE	
10001	104	Pilot	10010	Ramas	Alfred	A	615	844-2573	0.00	
10002	101	Pilot	10011	Dunne	Leona	K	713	894-1238	0.00	
10003	105	Pilot	10012	Smith	Kathy	w	615	894-2285	896.54	
10003	109	Copilot	10013	Olowski	Paul	F	615	894-2180	1285.19	
10004	106	Pilot	10014	Orlando	Myron		615	222-1672	673.21	
10005	101	Pilot	10015	O'Brian	Amy	B	713	442-3381	1014.56	
10006	109	Pilot	10016	Brown	James	G	615	297-1228	0.00	
10007	104	Pilot	10017	Williams	George		615	290-2556	0.00	
10007	105	Copilot	10018	Farriss	Anne	G	713	382-7185	0.00	
10008	106	Pilot	10019	Smith	Olette	K	615	297-3809	453.98	
10009	105	Pilot								
10010	108	Pilot								
10011	101	Pilot								
10011	104	Copilot								
10012	101	Pilot								
10013	105	Pilot								
10014	106	Pilot								
10015	101	Copilot								
10015	104	Pilot								
10016	105	Copilot								
10016	109	Pilot								
10017	101	Pilot								
10018	104	Copilot								
10018	105	Pilot								

  

Table name: EMPLOYEE						
EMP_NUM	EMP_TITLE	EMP_LNAME	EMP_FNAME	EMP_INITIAL	EMP_DOB	EMP_HIRE_DATE
100	Mr.	Kolmycz	George	D	15-Jun-1942	15-Mar-1987
101	Ms.	Lewis	Rhonda	G	19-Mar-1965	25-Apr-1988
102	Mr.	Vandam	Rhett		14-Nov-1958	20-Dec-1992
103	Ms.	Jones	Anne	M	16-Oct-1974	28-Aug-2005
104	Mr.	Lange	John	P	08-Nov-1971	20-Oct-1996
105	Mr.	Williams	Robert	D	14-Mar-1975	08-Jan-2006
106	Mrs.	Duzak	Jeanine	K	12-Feb-1968	05-Jan-1991
107	Mr.	Diante	Jorge	D	21-Aug-1974	02-Jul-1996
108	Mr.	vWesenbach	Paul	R	14-Feb-1966	18-Nov-1994
109	Ms.	Travis	Elizabeth	K	18-Jun-1961	14-Apr-1991
110	Mrs.	Genkazi	Leighla	w	19-May-1970	01-Dec-1992

  

Table name: AIRCRAFT					Table name: PILOT					
AC_NUMBER	MOD_CODE	AC_TTAF	AC_TTEL	AC_TTER	EMP_NUM	PIL_LICENSE	PIL_RATINGS	PIL_MED_TYPE	PIL_MED_DATE	PIL_PT135_DATE
1484P	PA23-250	1833.1	1833.1	101.8	101	ATP	ATP/SELMELInstr/CFI	1	20-Jan-08	11-Jan-08
2289L	C-90A	4243.8	768.9	1123.4	104	ATP	ATP/SELMELInstr	1	18-Dec-07	17-Jan-08
2778V	PA31-350	7992.9	1513.1	789.5	105	COM	COMM/SELMELInstr/CFI	2	05-Jan-08	02-Jan-08
4278Y	PA31-350	2147.3	622.1	243.2	106	COM	COMM/SELMELInstr	2	10-Dec-07	02-Feb-08
					109	COM	ATP/SELMEL/SESInstr/CFI	1	22-Jan-08	15-Jan-08

  

Table name: RATING		
RTG_CODE	RTG_NAME	
CFI	Certified Flight Instructor	
CFII	Certified Flight Instructor, Instrument	
INSTR	Instrument	
MEL	Multiengine Land	
SEL	Single Engine, Land	
SES	Single Engine, Sea	

  

Table name: MODEL				
MOD_CODE	MOD_MANUFACTURER	MOD_NAME	MOD_SEATS	MOD_CHG_MILE
C-90A	Beechcraft	KingAir	8	2.67
PA23-250	Piper	Aztec	6	1.93
PA31-350	Piper	Navajo Chieftain	10	2.35

Download the file named "assignment5\_database.sql" from D2L. Run the file from MySQL command window using the command (assuming the file is stored under c:\\):

```
mysql> source c:\\assignment5_database.sql
```

After the file completes, the tables shown above are created and populated with the shown data. Familiarize yourself with the database before starting to write your queries. Once you are familiar with the data, start working on the following questions.

**Answer the following questions based on the above database:**

- 1) Create a view named “ATL\_CHARTERS\_V” that includes the following “For each charter of aircraft to Atlanta (ATL), print the charter date, charter hours wait, and the corresponding customer last name, first name, area code, and phone number”. Your answer should include both the SQL statement for view creating along with the contents of the view (you get the contents of the view by select \* from ATL-CHARTERS\_V).

```
CREATE VIEW ATL_CHARTERS_V AS
(SELECT ch.char_date, ch.char_hours_wait, c.cus_lname, c.cus_fname, c.cus_areacode,
c.cus_phone
FROM Charter ch, Customer c
WHERE ch.cus_code = c.cus_code
AND char_destination = “ATL”);
```

```
mysql> select * from ATL_CHARTERS_V;
+-----+-----+-----+-----+-----+-----+
| char_date | char_hours_wait | cus_lname | cus_fname | cus_areacode | cus_phone |
+-----+-----+-----+-----+-----+-----+
| 2008-02-05 | 2.2 | Dunne | Leona | 713 | 894-1238 |
| 2008-02-06 | 3.5 | Dunne | Leona | 713 | 894-1238 |
| 2008-02-07 | 3.2 | Brown | James | 615 | 297-1228 |
| 2008-02-09 | 2.1 | Williams | George | 615 | 290-2556 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

- 2) Modify the MODEL table to add the following attribute and insert the values shown in the following table. (Note: use ALTER TABLE and UPDATE commands.)

Attribute name	Attribute Description	Attribute type	Attribute Values
MOD_LIFT_WEIGHT	Amount of weight each model can lift:	Numeri c	10,000 for C-90A 5,000 for PA23-250 20,000 for PA31-350

```
ALTER TABLE Model
ADD COLUMN MOD_LIFT_WEIGHT int;
```

```
UPDATE Model
SET mod_lift_weight = 10000
WHERE mod_code = “C-90A”;
```

```
UPDATE Model
SET mod_lift_weight = 5000
```

```
WHERE mod_code = "PA23-250";
```

```
UPDATE Model  
SET mod_lift_weight = 20000  
WHERE mod_code = "PA31-350";  
Commit;
```

- 3) Create a trigger named `trg_char_hours` that will automatically update the AIRCRAFT table when a new CHARTER row is added. Use the CHARTER table's CHAR\_HOURS\_FLOWN to update the AIRCRAFT table's AC\_TTAF, AC\_TTEL, and AC\_TTER values. The meaning for the AIRCRAFT table columns are as follows:  
**AC\_TTAF:** Total time on the air frame  
**AC\_TTEL:** Total time on the left engine (Also used to record single engine hours)  
**AC\_TTER:** Total time on the right engine.

So in the trigger, you need to increase all of them with CHAR\_HOURS\_FLOWN

```
delimiter //  
CREATE TRIGGER trg_char_hours  
AFTER INSERT ON Charter  
FOR EACH ROW  
BEGIN  
    UPDATE Aircraft  
    SET ac_ttaf = ac_ttaf + new.char_hours_flowndata-bbox="171 578 857 662" data-label="List-Group">

4) Create a trigger named trg_cust_balance that will automatically update the CUSTOMER table's CUS_BALANCE when a new CHARTER row is added. Use the CHARTER table's CHAR_TOT_CHG as the update source (Assume that all charter charges are charged to the customer balance.) In addition to the CHAR_TOT_CHG, add $5 for every quart of oil used on the charter.


```

```
delimiter //  
CREATE TRIGGER trg_cust_balance  
AFTER INSERT ON Charter  
FOR EACH ROW  
BEGIN  
    UPDATE Customer  
    SET cus_balance = cus_balance + new.char_tot_chg + (5 *  
new.char_oil_qts)  
    WHERE Customer.cus_code = new.cus_code;  
END;  
//
```

- 5) Create a stored procedure to update model charge per mile attribute. Procedure takes the model number as a parameter. The procedure increases the charge for this model by 20%.

```
CREATE PROCEDURE prc_mod_charge_increase (IN W_MOD_CD varchar(10))
BEGIN
    UPDATE    Model
    SET       mod_chg_mile = mod_chg_mile * 1.2
    WHERE     mod_code = W_MOD_CD;
END
//
```

- 6) Create a stored procedure that will take an Employee number and percentage, then update the corresponding employee's hourly salary by the input percentage (increase the hourly salary, so you are giving the employee a raise). Hint: you may have to do some additional work on this one to get the table correct.

```
alter table employee
add hourly_salary double;

update employee
set hourly_salary = 20;

commit;

delimiter //
create procedure prc_increase_salary(in W_EMPLOYEE
int(11), in W_PERCENT_INCREASE double)
begin
    update employee
        set hourly_salary = hourly_salary * (1 +
        W_PERCENT_INCREASE / 100)
        where emp_num = W_EMPLOYEE;
end//
delimiter ;
```