

1. Was February 30 ever used in some country's calendar? When, where and why did that happen?

Yes, 30th February was indeed used in the Swedish calendar in 1712. In 1700 Sweden, the Empire planned to change the Julian calendar to the Gregorian calendar. Therefore the leap day was missed in February 1700. Moreover, the war diverted people's attention so leap day in 1704 and 1708 were not missed. To avoid confusion and further mistakes, the Julian calendar was restored in 1712 by adding an extra leap day. February 30 1712 came into existence in Sweden when the Julian calendar was restored and two leap days were added that year.

2.

With the actual solar year being 365.24219878 days, which of these three calendars is the most accurate:

- a. Julian,
- b. Gregorian, or
- c. Jalaali (8 leaps every 33 years)?

Number of days in a year in Julian Calendar: $(4 \times 365 + 1)/4 = 365.25$
1 leap year in every 4 years

In Gregorian Calendar: $(400 \times 365 + 97)/400 = 365.2425$
97 leap years every 400 years.

In a year in Jalali Calendar: $(33 \times 365 + 8)/33 = 365.2424$
8 leap years every 33 years

Therefore, Jalali Calendar are more close to the actual number of days in a year.

3. Is "honey moon" a celestial body or a month--and what is origin of the expression?

The meaning of the word "Honeymoon" can be traced back to the fifth century. Cultures represented calendar time with moon cycles: one rotation of moon around the earth was one month. At that time, a newlywed couple drank mead during their first month of marriage. "Honeymoon" was thus, actually the month after a wedding. The Babylonians started calling the month the "honey month" and the Babylon calendar was a lunar calendar. However, now it is called as "honeymoon".

4. Do exercises 6.1, 6.3 from the ADS textbook. (Errata: for Exercise 6.3, the current tuple contains an additional Dosage attribute with value 100).

6.1

- a. Define the Employee relation as a bitemporal table using TSQL2's CREATE TABLE statement.
- b. Express the following in TSQL2 on this relation:
 - i. What is the history of the maximum salary?
 - ii. What is the history of the average salary?

- a. CREATE TABLE Employee(Name CHAR(30), Salary INT, Title CHAR(30),DateOfBirth DATE)
AS VALID STATE DAY AND TRANSACTION
- b1 SELECT MAX(Salary) FROM Employee
- b2 SELECT AVG(Salary) FROM Employee

6.3

Name	Drug	Dosage	Valid Time	Transaction Time
Melanie	Proventil	100	[1996-01-01 - 1996-08-31]	[1996-06-01 – 1996-09-15]
Melanie	Proventil	100	[1996-01-01 - 1996-02-29]	[1996-09-15 – until changed]
Melanie	Proventil	50	[1996-03-01 – 1996-05-30]	[1996-09-15 – until changed]
Melanie	Proventil	100	[1996-06-01 – 1996-08-31]	[1996-09-15 – until changed]

5. Give a simpler SQL expression for the temporal joins of ADS example 5.9.

```
SELECT DISTINCT E1.Name, E1.Salary, E2.Title,
CASE WHEN E1.Start < E2.Start THEN E2.Start ELSE E1.Start END
CASE WHEN E1.Stop > E2.Stop THEN E2.Stop ELSE E1.Stop END
FROM Employee1 E1, Employee2 E2
WHERE E1.Name = E2.Name AND E1.Start < E2.Stop AND E1.Stop > E2.Start
```

6. Write an test on a Datalog system rules to coalesce the periods after Sal is projected out from EHist(Eno, Sal, Title, From, To)

Coalesce.fac

```
% ehist(Name:string, Salary:integer, Title:string, From:integer, To:integer)
ehist('A', 60000, 'Assistant Provost', 19950101, 19950601).
ehist('A', 70000, 'Assistant Provost', 19950601, 19951001).
ehist('A', 70000, 'Provost', 19951001, 19960201).
ehist('A', 70000, 'Professor', 19960201, 19970101).
```

Coalesce.deal

```
database(ehist(Name:string,Salary:integer,Title:string,From:integer,To:integer)).
```

```
getNameSalary (Name,Salary) <- ehist(Name,Salary,_,_,_).
export getNameSalary(N,S).
```

```
getDatesforSalary (Name,Salary, From, To) <- getNameSalary(Name,Salary),
ehist(Name,Salary,_,From,To).
```

```
export getDatesforSalary(N,S,F,T).
```

```
coalesceSal (Name,Salary,From,To) <- getDatesforSalary(Name,Salary,From,To),
~getDatesforSalary(Name,Salary,To,To2).
```

```
coalesceSal (Name,Salary,From1,To2) <- getDatesforSalary(Name,Salary,From1,To1),
coalesceSal(Name,Salary,To1,To2).
```

```
export coalesceSal(N,S,F,T).
```

```
removeExtra (Name,Salary,From2,To) <- coalesceSal(Name,Salary,From1,To),
coalesceSal(Name,Salary,From2,To), From2 > From1.
```

```
export removeExtra(N,S,F,T).
```

```
finalOutput (Name,Salary,From,To) <- coalesceSal(Name,Salary,From,To),
~removeExtra(Name,Salary,From,To).
export finalOutput(N,S,F,T).
```

7.

```
db2 CREATE TABLE table.ehist (Eno int, Salary int, from date, to date)
```

```
db2 INSERT INTO table.ehist VALUES('A', 60000, '01/01/1995', '06/01/1995');
db2 INSERT INTO table.ehist VALUES('A', 70000, '06/01/1995', '10/01/1995');
db2 INSERT INTO table.ehist VALUES('A', 70000, '10/01/1995', '02/01/1996');
db2 INSERT INTO table.ehist VALUES('A', 70000, '02/01/1996', '01/01/1997');
```

```
with coal(Name, Salary, From, To, Count) As ((select Eno, Salary, from, to, 0 from table.ehist)
union all
(select coal.Name, coal.Salary, From, to, Count+1
from coal, table.ehist where Count<5 and table.ehist.eno = coal.Name and table.ehist.Salary=coal.Salary and
From <= from and from <= To and To < to))
```

```
select distinct * from coal;
with coal(Name, Salary, From, To, Count) As
((select Eno, Salary, from, to, 0 from table.ehist)
union all
(select coal.Name, coal.Salary, From, to, Count+1
from coal, table.ehist
where Count<5 and table.ehist.eno = coal.Name and table.ehist.Salary=coal.Salary and
From <= from and from <= To and To < to))
```

```
select distinct * from coal
where not exists
(select * from coal as C
where
C.eno=coal.Name and C.Salary=coal.Salary
and C. From <= Coal. From and C.To>=coal. To
and (C. From <Coal. From or C. To >coal. To));
```

8.

```
CREATE TRIGGER delete_emp
AFTER DELETE ON EMP
FOR EACH ROW
UPDATE EHIST AS E
SET E.To = CURRENT_DATE
WHERE E.Sal = OLD.Sal AND E.Title = OLD.Title AND E.Eno = OLD.Eno
```

```
CREATE TRIGGER insert_emp
AFTER INSERT ON EMP
FOR EACH ROW
INSERT INTO EHist (Eno, Sal, Title, From, To)
VALUES(NEW.Eno, NEW.Sal, NEW.Title, CURRENT_DATE, NULL)
```

```
CREATE TRIGGER update_emp
AFTER UPDATE ON EMP
FOR EACH ROW
BEGIN
UPDATE EHist
SET To=CURRENT_DATE
WHERE To=NULL AND Eno=NEW.Eno AND Sal = OLD.Sal AND Title = OLD.Title
INSERT INTO EHist (Eno, Sal, Title, From, To)
VALUES (NEW.Eno, NEW.Sal, NEW.Title, CURRENT_DATE, NULL)
END
```