CS240A: Homework 4

Author: Sneha Shankar
UID: 404946026

1.a) Yes, 30th February was indeed used in the Swedish calendar in 1712! The story is:
In 1700 Sweden, planned to convert from the Julian calendar to the Gregorian calendar. Therefore 1700, which should have been a leap year in the Julian calendar, was not a leap year in Sweden. However, 1704 and 1708 became leap years by error in Sweden's calendar. This left Sweden out of synchronization with both the Julian and the Gregorian calendars, so the country reverted back to the Julian calendar. February 30 1712 came into existence in Sweden when the Julian calendar was restored and two leap days were added that year [1].

1.b) The meaning of the word "Honeymoon" can be traced back to the fifth century. This is the time when cultures represented calendar time with moon cycles i.e. one rotation of moon around the earth was one month. Back then, a newlywed couple drank mead (honey-based alcoholic drink) during their first month (month was called as 'moon') 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" [2].

1.c) The **actual number of days** in a year is 365.24219878 and not 365.

Number of days in a year in **Julian Calendar**: (4*365 + 1)/4 = 365.25 i.e. 1 leap year in every 4 years

Number of days in a year in **Gregorian Calendar**: (400*365 + 97)/400 = 365.2425 i.e. 97 leap years every 400 years.

Number of days in a year in **Jalali Calendar**: (33*365 + 8)/33 = 365.2424 i.e. 8 leap years every 33 years

As stated above, we see that the number of days in Jalali Calendar are more close to the actual number of days in a year. Hence, we can say that the Jalali Calendar is the most accurate calendar.

2) 6.1)

a) **CREATE TABLE** Employee(Name **CHAR(30),** Salary **INT**, Title **CHAR(30)**, DateOfBirth **DATE**) **AS VALID STATE DAY AND TRANSACTION**

b) i) **SELECT MAX(**Salary**) FROM** Employee

b) ii) **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	[1996-05-31 − 1996-08-31]	[1996-09-15 − until changed]

3) 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

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

getStats(Name,Salary,From,To)<-ehist(Name,Salary,_,From,To).
export getStats(N,S,F,T).

smaller(From1,From2,From1) <- From1<=From2.
smaller(From1,From2,From2) <- From1>From2.

larger(To1,To2,To1) <- To1>=To2.
larger(To1,To2,To2) <- To2>To1.

export smaller($X,$Y,Z).
export larger($X,$Y,Z).

coalesce(Name,Salary,From,To) <- getStats(Name,Salary,From,To).
coalesce(Name,Salary,From,To) <- coalesce(Name,Salary,From1,To1),
getStats(Name,Salary,From2,To2), From1<=To2, From2<=To1, smaller(From1,From2,From),
larger(To1,To2,To).
export coalesce(N,S,F,T).

isLarger(Name,Salary,From,To) <- coalesce(Name,Salary,From1,To1), From1<From, To1>=To.
isLarger(Name,Salary,From,To) <- coalesce(Name,Salary,From1,To1), From1<=From, To1>To.
export isLarger($N,$S,$F,$T).

finalCoalesce(Name,Salary,From,To)<-coalesce(Name,Salary,From,To),
~isLarger(Name,Salary,From,To).

```
export finalCoalesce(N,S,F,T).
query finalCoalesce(N,S,F,T).
```

```
root@LAPTOP-POR90PGJ:/mnt/c/Sneha/Studies/UCLA/
le.sh DeALS-0.9.jar Q4_HW4.deal
Executing query 'finalCoalesce(N, S, F, T).'.
Query Id = finalCoalesce(N,S,F,T).1525492515359
Query Form = finalCoalesce(N, S, F, T).
Return Status = SUCCESS
Execution time = 12ms
finalCoalesce(Bob, 60000, 19950101, 19950601).
finalCoalesce(Bob, 70000, 19950601, 19970101).
```

References:
1. https://en.wikipedia.org/wiki/Portal:Organized_Labour/February/30
2. https://timesofindia.indiatimes.com/Why-is-a-honeymoon-called-so/articleshow/8502866.cms