Работа с датами и временем

Содержание

Временные шкалы

В идеале системы учёта времени должны обладать тремя характеристиками:

1. *точность* (время базируется на атомном стандарте, каждая секунда отсчитывается как секунда в системе СИ, нет високосных секунд, переводов на зимнее или летнее время и т.п.);
2. *простота* (каждый ‘день’ состоит из 86400 “секунд”);
3. *календарные дни* (дни календаря точно соответствуют вращению Земли).

На практике есть возможность выбрать шкалу только с двумя характеристиками из трёх.

1. точность и календарные дни. Примером такой шкалы является UTC, в которой отсчёт дней и секунд ведётся разными методами (секунды исчисляются по атомному стандарту, а дни по суточному вращению Земли), а соответствие достигается вводом *секунды координации*;
2. календарные дни и простота. Примером такой шкалы является POSIX (IEEE Std 1003.1-1988), в которой день всегда равен 86400 секундам;
3. точность и простота. Этими характеристиками обладают технические шкалы (атомные часы, GPS), в которых не важен учёт дней.

Классы времени

Время можно условно поделить на два класса: физическое и гражданское.

1. *Физическое* время представляет собой точки на непрерывной шкале, такую концепцию достаточно точно отражает UTC, если можно пренебречь *секундой координации* (дополнительная секунда, добавляемая к UTC 30 июня или 31 декабря для согласования со средним солнечным временем UT1. В этот момент время условно обозначается как

23:59:60, а на шкале UTC две секунды отображаются как одна).

2. *Гражданское* время представляется полями (год, месяц, число, час, минута, секунда, доли секунды, а также временная зона и календарь, который по умолчанию считается Григорианским).

Почти всегда существует возможность однозначно перевести физическое время в гражданское и наоборот, но при этом следует различать их свойства и применение (ближайшая аналогия — массивы байтов и символьные строки).

Основная проблема состоит в том, что в обычном употреблении не всегда можно сделать вывод о том, какой класс времени подразумевается, и гражданское время меняется на основании юридических актов (указов, постановлений и т.п.), что приводит к неоднозначности при планировании событий.

Обработка на компьютере

Хранение и отображение

Для ссылки на момент во времени необходимо использовать единую шкалу, на которой нет разрывов, связанных с летним временем. Примером стандарта времени с такой шкалой является UTC. Локальное или местное время для таких целей не подходит, так как на его шкале имеются разрывы и неоднозначности, связанные с переходами на летнее время и обратно.

Если нужно сохранить значение локального времени, то необходимо сохранить также смещение относительно UTC, чтобы временную отметку можно было интерпретировать однозначно. Необходимо помнить, что местное время может отличаться от UTC на интервал не кратный часу (например, в Нидерландах с 1909-05-01 по 1937-06-30 смещение времени от UTC составляло 19 минут и 32.13 секунд).

Правила хранения и отображения времени:

1. Время всегда хранится в UTC. При необходимости дополнительно сохраняется информация о временной зоне (смещение и/или название).
2. Для вывода на экран время переводится в местное.
3. При выводе на экран отличного от местного времени необходимо указывать название часового пояса или типа исчисления (например, Юлианский календарь).

Системное время для администратора

Системный администратор должен следить за тем, чтобы файлы базы данных описания временных зон **tzdata** имели одну версию в рамках комплекса и обновлялись регулярно, чтобы минимизировать расхождения с внешними системами. Особенно это важно делать при издании новых правил учёта зимнего или летнего времени.

На серверах аппаратные часы всегда должны использовать UTC. Операционные системы

тоже должны использовать UTC, чтобы при возникновении проблем в файлах журналов было указано время в едином формате.

В рамках комплекса следует использовать синхронизацию времени, даже если нет доверенного источника более высокого уровня.

Клиентские приложения

В общем случае нельзя доверять временным отметкам внешних клиентов, так как невозможно гарантировать их корректность. Программа должна самостоятельно ставить временные отметки для происходящих событий. Исключением являются программные комплексы, в которых предусмотрено наличие доверенных приложений, ответственных за установку временных отметок.

Postgresql

Сервер базы данных должен использовать только временную зону UTC. Для этого в файле настройки сервера `postgresql.conf` должна быть строка

```
timezone = 'UTC'
```

При создании таблиц необходимо использовать типы данных `timestamp without time zone` и `time without time zone`. Если данные о времени должны содержать информацию о временной зоне или смещении относительно UTC, то нужно создать дополнительные столбцы, информацию из которых должно обрабатывать клиентское приложение. Например, временные отметки можно хранить в таком виде

```
CREATE TABLE time_stamps (
    time_stamp time without time zone, -- временная отметка
    time_zone character(12),           -- текстовое название временной зоны
    shift integer                      -- смещение локального времени относительно UTC
в секундах
);
```

Смещение следует записывать в четырёхбайтное знаковое целое, так как оно может быть как положительным, так и отрицательным, а его максимальное значение может составлять 14 * 60 * 60 = 50400 секунд. Несмотря на приведённый выше пример с временной зоной Нидерландов, микросекундами можно пренебречь, так как современные временные зоны имеют смещения кратные минутам, а точность до микросекунд в далёком прошлом обычно не требуется.

Чтобы клиентская программа могла получить выборку, привязанную к некоторой временной зоне можно использовать оператор `AT TIME ZONE`, например

```
SELECT TIMESTAMP '2001-02-16 20:38:40Z' AT TIME ZONE 'MSK';
```

В результате время будет преобразовано из временной зоны UTC в MSK. Модификатор `Z` в записи времени указывает на принадлежность зоне UTC.

Qt

Класс `QTimeZone` предназначен для получения информации о временной зоне и перевода времени из одной временной зоны в другую.