Dates in SQL

SQL provides support for several data types to represent dates and time intervals, and several functions for them, although there may be slight differences in implementations between DBMSs.

The most useful data type is probably DATE, which represents a date. There is also TIMESTAMP, which represents a date and a time, and INTERVAL which represents a time interval. There are also several variations on these types, having different granularities (say going to the second or millisecond etc) or dealing with time zones.

```
CREATE TABLE Task (
     Id Integer PRIMARY KEY,
     Description VARCHAR(20),
     DueDate DATE,
     StartTime TIMESTAMP
);
```

Example 1: CREATE TABLE statement using date and timestamp. Notice that the TIMESTAMP type includes BOTH a date and a time.

Date Constants

SQL automatically converts from a string to a date constant; with the only issue being the actual representation of the date as a string, which is dependent not just on the DBMS but also on the *locale*, that is, the local rules (in the USA, for example, we do dates as month-day-year, while in Europe most people do day-month-year, plus we get into issues of what character to use as a separator, ie, - vs /, and whether to represent the month as a number or with letters).

In general, it is *usually* safe to use the 3-letter abbreviation for a month, and to use – to separate the dates. One notation that works for both Oracle and PostgreSQL in our local installation is `'12-jan-1972'`. For TIMESTAMP constants, we can use a date, then a space, and then the time. We can use a colon (:) to separate the time components; Oracle requires that we specify hours, minutes and seconds with this notation.

For example, we could insert rows in our task table as follows:

```
INSERT INTO Task(Id,Description,DueDate,StartTime) VALUES
     (1,'Do something','3-feb-2009','24-Oct-2008 3:45:00 pm');
INSERT INTO Task(Id,Description,DueDate,StartTime) VALUES
     (2,'Do stuff','3/feb/2009','24-Sep-2008');
```

Example 2: CREATE TABLE statement using date and timestamp. Notice that the TIMESTAMP type includes BOTH a date and a time.

Unfortunately, the data would NOT be interpreted consistently; In the second row, since we're not specifying a time, each DBMS could do different things (in our local installations, Oracle would use 8 am as the default time, whereas postgresql would use 0 am).

Operations on Dates

- We can subtract two dates by using the minus (-) operator, and we get the number of days in between those dates.
- If we subtract two timestamps, or a timestamp and a date, we get an *interval*, which represents the amount of time between dates.
- We can use the EXTRACT function to get the individual fields (day, month etc) from a date, timestamp or interval value. For example, we could use EXTRACT(Year FROM dueDate) to get only the year part of the due date.
- Both Oracle and PostgreSQL provide pseudo-constants named CURRENT_DATE and CURRENT_TIMESTAMP that yield the current date and time.
- Both Oracle and PostgreSQL provide a function to_date that converts a string to a date given a template; they also provide a to_timestamp function, and a to_char function to format dates.