

Recitation -2-

Structured Query Language SQL - DDL

- ◆ SQL Datatypes
- ◆ DDL statements

Presented by:
Rakan A. Alseghayer

Basic SQL-DDL COMMANDS

- ❑ For database schemas:

CREATE SCHEMA, DROP SCHEMA

- ❑ For tables:

CREATE TABLE, DROP TABLE, ALTER TABLE

- ❑ For views:

CREATE VIEW, DROP VIEW

- ❑ For integrity constraints

CREATE IC, DROP IC

- ❑ For domains:

CREATE DOMAIN, DROP DOMAIN [SQL99]

For Indexes [defunct in SQL2]

Database Schema

- ❑ **CREATE SCHEMA** <database-name>
 AUTHORIZATION <user-identifier>;
- ❑ E.g. **CREATE SCHEMA** micro_db
 AUTHORIZATION panos;
- ❑ **DROP SCHEMA** <db-name> [**RESTRICT** | **CASCADE**];
 - Restrict: removes the schema if the db has no data
 - Cascade: removes everything, data and definitions
- ❑ E.g., **DROP SCHEMA** micro_db **RESTRICT**;

Schema and Catalog

- ❑ SQL2, SQL3 support multiple database schemas
- ❑ **Catalog** contains the definitions of database schemas
- ❑ INFORMATION_SCHEMA
 - Schemas and Base relations (tables)
(tbl_name, creator, #of_tuples, tuple_length, #of_attributes...)
 - Attributes of Relations (columns)
(tbl_name, atrb_name, type, format, order, key_no, ...)
 - Authorization
 - Integrity
 - Indexes
- ❑ Naming of tables: Schema_name.Table_name
- ❑ Query: Describe table name; or using SELECT

Create Table

❑ **CREATE Table** <Table-name> (
 <Attribute-name> <Attribute-Type>, ...
 Constraint <Constraint-name> <Constraint-spec>, ...);

❑ E.g.,

```
CREATE TABLE Students (  
    sid CHAR (20) ,  
    name CHAR (20) ,  
    psid INTEGER,  
    age INTEGER,  
    gpa REAL,  
    timestamp  
    Constraint Student_PK  
    PRIMARY KEY (sid) );
```

SQL Datatypes

- ❑ Numeric
 - Fixed numbers, approximate numbers, formatted numbers
- ❑ Character Strings
 - fixed & varying length, CLOBS [SQL99], foreign language
- ❑ Bit Strings
 - fixed & varying length, BLOBS [SQL99]
- ❑ Temporal Data
 - date, time and timestamp, intervals
- ❑ **NULL** value valid for all types

SQL Character Strings

- ❑ Concatenation operator: ||
 - 'abc' || 'XYZ' results in 'abcXYZ'

- ❑ Foreign-language characters (ISO-defined chars):
 - NATIONAL CHAR(n)
 - NATIONAL VARCHAR(n)

SQL Temporal Data

- ❑ DATE data type
- ❑ TIME and TIMESTAMP data types
- ❑ INTERVAL data type.
 - INTERVAL data type represents periods of time

Date and Time

- ❑ **DATE** (10 positions) stores calendar values representing YEAR, MONTH, and DAY: **YYYY-MM-DD**
- ❑ **TIME** defines HOURS, MINUTES, and SECONDS in a twenty-four-hour notation: **HH:MM:SS**
- ❑ **TIME(i)** defines *i* additional decimal fractions of seconds: **HH:MM:SS:ddd...d**
- ❑ **TIME WITH TIME ZONE** includes the displacement [+13:00 to -12:59] from standard universal time zone: **HH:MM:SS{+/-}hh:mm**
 - *hh* are the two digits for the TIMEZONE_HOUR and *mm* the two digits for TIMEZONE_MINUTE
- ❑ **TIMESTAMP** represents a complete date and time with 6 fractions of seconds and optional time zone.

Functions on Dates

- ❑ All systems provide functions under different names
 - for constructing a date from strings or integers
 - for extracting out the month, day, or year from a date
 - for displaying dates in different ways
- ❑ Examples,
 - CAST(string AS DATE) [SQL2: CAST(<value> AS <type>)]
e.g., CAST('2002-02-18' AS DATE)
 - MAKEDATE (int year, int month, int day) or
DATE (int year, int month, int day)
e.g., MAKEDATE(1999, 12, 31)
 - EXTRACT (MONTH/DAY/YEAR FROM <date>) [SQL3]
 - YEAR(<date>), MONTH(<date>), DAY(<date>)

PostgreSQL Functions on Dates

Function	Return Type	Description	Example	Result
age(timestamp, timestamp)	interval	Subtract arguments, producing a "symbolic" result that uses years and months	age(timestamp '2001-04-10', timestamp '1957-06-13')	43 years 9 mons 27 days
age(timestamp)	interval	Subtract from current_date (at midnight)	age(timestamp '1957-06-13')	43 years 8 mons 3 days
clock_timestamp()	timestamp with time zone	Current date and time (changes during statement execution); see Section 9.9.4		
current_date	date	Current date; see Section 9.9.4		
current_time	time with time zone	Current time of day; see Section 9.9.4		
current_timestamp	timestamp with time zone	Current date and time (start of current transaction); see Section 9.9.4		
date_part(text, timestamp)	double precision	Get subfield (equivalent to extract); see Section 9.9.1	date_part('hour', timestamp '2001-02-16 20:38:40')	20
date_part(text, interval)	double precision	Get subfield (equivalent to extract); see Section 9.9.1	date_part('month', interval '2 years 3 months')	3
date_trunc(text, timestamp)	timestamp	Truncate to specified precision; see also Section 9.9.2	date_trunc('hour', timestamp '2001-02-16 20:38:40')	2001-02-16 20:00:00
extract(field from timestamp)	double precision	Get subfield; see Section 9.9.1	extract(hour from timestamp '2001-02-16 20:38:40')	20
extract(field from interval)	double precision	Get subfield; see Section 9.9.1	extract(month from interval '2 years 3 months')	3
isfinite(date)	boolean	Test for finite date (not +/-infinity)	isfinite(date '2001-02-16')	true
isfinite(timestamp)	boolean	Test for finite time stamp (not +/-infinity)	isfinite(timestamp '2001-02-16 21:28:30')	true
isfinite(interval)	boolean	Test for finite interval	isfinite(interval '4 hours')	true
justify_days(interval)	interval	Adjust interval so 30-day time periods are represented as months	justify_days(interval '35 days')	1 mon 5 days
justify_hours(interval)	interval	Adjust interval so 24-hour time periods are represented as days	justify_hours(interval '27 hours')	1 day 03:00:00
justify_interval(interval)	interval	Adjust interval using justify_days and justify_hours, with additional sign adjustments	justify_interval(interval '1 mon -1 hour')	29 days 23:00:00
localtime	time	Current time of day; see Section 9.9.4		
localtimestamp	timestamp	Current date and time (start of current transaction); see Section 9.9.4		
now()	timestamp with time zone	Current date and time (start of current transaction); see Section 9.9.4		
statement_timestamp()	timestamp with time zone	Current date and time (start of current statement); see Section 9.9.4		
timeofday()	text	Current date and time (like clock_timestamp, but as a text string); see Section 9.9.4		
transaction_timestamp()	timestamp with time zone	Current date and time (start of current transaction); see Section 9.9.4		

Operations on Dates

- ❑ Datetime (+ or -) Interval = Datetime
- ❑ Datetime - Datetime = Interval
- ❑ Interval (* or /) Number = Interval
- ❑ Interval (+ or -) Interval = Interval

- ❑ Examples (ANSI SQL):
 - (CURRENT_DATE + INTERVAL '1' MONTH)
 - (CURRENT_DATE - INTERVAL '18' DAY)
 - (CURRENT_DATE - BirthDate)

Intervals

- ❑ An interval results when two dates are subtracted.
E.g., AdmitDate – DischargeDate
- ❑ Two interval data types: **Year-Month** & **Day-Time**
- ❑ Format: INTERVAL start-field(p) [TO end-field(fs)]
 - p is the precision (default is 2 digits)
 - fs is the fractional second precision, which is only applicable to DAY/TIME (default is 6 digits)
- ❑ Year-Month intervals:
 - INTERVAL YEAR, INTERVAL YEAR(p), INTERVAL MONTH, INTERVAL MONTH(p), INTERVAL YEAR TO MONTH, INTERVAL YEAR(p) TO MONTH
 - E.g., INTERVAL YEAR (2) to MONTH could be [0-0, 99-11]

PostgreSQL Intervals...

Operator	Example	Result
+	date '2001-09-28' + integer '7'	date '2001-10-05'
+	date '2001-09-28' + interval '1 hour'	timestamp '2001-09-28 01:00:00'
+	date '2001-09-28' + time '03:00'	timestamp '2001-09-28 03:00:00'
+	interval '1 day' + interval '1 hour'	interval '1 day 01:00:00'
+	timestamp '2001-09-28 01:00' + interval '23 hours'	timestamp '2001-09-29 00:00:00'
+	time '01:00' + interval '3 hours'	time '04:00:00'
-	- interval '23 hours'	interval '-23:00:00'
-	date '2001-10-01' - date '2001-09-28'	integer '3' (days)
-	date '2001-10-01' - integer '7'	date '2001-09-24'
-	date '2001-09-28' - interval '1 hour'	timestamp '2001-09-27 23:00:00'
-	time '05:00' - time '03:00'	interval '02:00:00'
-	time '05:00' - interval '2 hours'	time '03:00:00'
-	timestamp '2001-09-28 23:00' - interval '23 hours'	timestamp '2001-09-28 00:00:00'
-	interval '1 day' - interval '1 hour'	interval '1 day -01:00:00'
-	timestamp '2001-09-29 03:00' - timestamp '2001-09-27 12:00'	interval '1 day 15:00:00'
*	900 * interval '1 second'	interval '00:15:00'
*	21 * interval '1 day'	interval '21 days'
*	double precision '3.5' * interval '1 hour'	interval '03:30:00'
/	interval '1 hour' / double precision '1.5'	interval '00:40:00'