CSc 484
Database Management Systems

Ken Gamradt

Spring 2024

Data Types

Data types

- Basic built-in data types
  - **char**(*n*)
  - **varchar**(*n*)
  - **int - integer**
  - **smallint**
  - **numeric**(*p*, *d*)
  - **real, double precision**
  - **float**(*n*)

Data types – Boolean

- Consists of the distinct values **TRUE** and **FALSE**
- Also supports the the **NULL** value
  - Unless prohibited by a **NOT NULL** constraint
- All Boolean data type values are mutually comparable and assignable
  - **TRUE** is greater than **FALSE**
  - Any comparison involving **NULL** or **UNKNOWN** returns an **UNKNOWN** result

Data types – Boolean

- SQL Server
  - No **boolean** data type
  - **bit**: integer that can be 0, 1, or NULL

```
create table degree(
    ID          varchar(5),
    phd         bit);

insert into degree
    values('1001', 1);

insert into degree
    values('1002', 0);

insert into degree
    values('1003', null);
```

	ID	phd
1	1001	1
2	1002	0
3	1003	NULL

Data types – Boolean

- MySQL
  - Supports **boolean** data type
  - Value can be: TRUE, FALSE, NULL, 1, 0

```
create table degree(
    ID          varchar(5),
    phd         boolean);

insert into degree
    values('1001', true);

insert into degree
    values('1002', false);

insert into degree
    values('1003', null);

insert into degree
    values('1004', 1);

insert into degree
    values('1005', 0);
```

ID	phd
1001	1
1002	0
1003	(NULL)
1004	1
1005	0

Data types – Boolean

- PostgreSQL
  - Supports **boolean** data type
  - Value can be: TRUE, FALSE, NULL

```
create table degree(
    ID          varchar(5),
    phd         boolean);

insert into degree
    values('1001', true);

insert into degree
    values('1002', false);

insert into degree
    values('1003', null);

insert into degree
    values('1004', 1);

insert into degree
    values('1005', 0);
```

	id character varying (5)	phd boolean
1	1001	true
2	1002	false
3	1003	[null]
4	1004	true
5	1005	false

Data types – Data and Time

- SQL standard supports several data types relating to dates and times
  - **DATE**: calendar date containing a (four-digit) year, month, and day of the month
  - **TIME**: time of day, in hours, minutes, and seconds
  - **TIMESTAMP**: combination of DATE and TIME

Data types – Data and Time

- DATE: must be specified in the format year followed by month followed by day
  - YYYY-MM-DD
  - '2022-01-03'

```
create table staff(
    ID          varchar(6),
    name        varchar(15),
    join_date   date);

insert into staff
    values( '10001', 'Gamradt', '1991-09-01');

select *
    from staff
    where join_date = '1991-09-01';
```

Data types – Data and Time

- TIME: can be specified in this format
  - HH:MM:SS
  - '09:30:00'

```
create table classes(
    ID          varchar(5),
    title       varchar(20),
    start_time  time); -- time(6)

insert into classes
    values ('001', 'Database', '9:30:00');

insert into classes
    values ('002', 'OOP', '12:00:00');

select *
    from classes;
```

	ID	title	start_time
1	001	Database	09:30:00.000000
2	002	OOP	12:00:00.000000

Data types – Data and Time

- TIME(p) can be used to specify the number of fractional digits for seconds
  - Default is 6

```
create table classes(
    ID          varchar(5),
    title       varchar(20),
    start_time  time(0));

insert into classes
    values ('001', 'Database', '9:30:00');

insert into classes
    values ('002', 'OOP', '12:00:00');

select *
    from classes;
```

	ID	title	start_time
1	001	Database	09:30:00
2	002	OOP	12:00:00

Data types – Data and Time

- TIMESTAMP: a combination of DATE and TIME
  - TIMESTAMP(p): p can be used to specify the number of fractional digits for seconds
    - default is 6
    - YYYY-MM-DD HH:MM:SS 2022-01-11 09:30:00

```
create table classes(
    ID          varchar(5),
    title       varchar(20),
    start_time  timestamp(0));  -- default is 6
    -- SQL Server prefers DATETIME or DATETIME2

insert into classes
    values ('001', 'Database', '2022-01-11 9:30:00');

select *
    from classes;
```

	id character varying (5)	title character varying (20)	start_time timestamp without time zone
1	001	Database	2022-01-11 09:30:00

Data types – Data and Time

- DATETIME in SQL Server

```
create table classes2(
    ID          varchar(5),
    title       varchar(20),
    start_time  datetime);

insert into classes2
    values ('001', 'Database', CURRENT_TIMESTAMP);
    -- returns datatime value including date and time
insert into classes2
    values ('002', 'OOP', '2023-01-09 12:00:00');
```

	ID	title	start_time
1	001	Database	2022-01-11 09:30:00.000
2	002	OOP	2022-01-10 12:00:00.000

Data types – Data and Time

- GETDATE() and SYSDATETIME(), for the current date and time
  - Postgres, NOW()

- SQL Server, GetUTCDate() gives current timestamp at GMT

Data types – Data and Time

- SQL defines several functions to get the current date and time
  - CURRENT_TIMESTAMP returns current date and time (with time zone)
  - LOCALTIMESTAMP returns current date and time (with time zone)
  - CURRENT_DATE returns current date
  - CURRENT_TIME returns the current time (with time zone)
  - LOCALTIME returns the current time (without time zone)
  - EXTRACT(field FROM d)
    - d: a date or time value
    - field: can be YEAR, MONTH, DAY, HOUR, MINUTE, or SECOND

- Not supported by all dialects
- Dialects may provide different names on those functions, or other extension

Generating Unique Key Values

- instructor(ID, name, dept_name, salary)
- ID, holds values created by the enterprise solely for ***identification purposes***

```
create table instructor(
    ID          varchar(5), -- generate unique values yourself
    name        varchar(20) not null,
    dept_name   varchar(20),
    salary      numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department (dept_name)
        on delete set null);
```

Generating Unique Key Values

- Database systems offer automatic management of unique-key generation
- Syntax differs among different database systems

Generating Unique Key Values

- SQL Server

IDENTITY [(seed, increment)]

- **Seed**: is the value that is used for the very first row loaded into the table
- **Increment**: is the incremental value that is added to the identity value of the previous row that was loaded
- You must specify both the seed and increment or neither
  - Default is (1, 1)

Generating Unique Key Values

- SQL Server

```
create table instructor(
    ID          int identity (10001, 1),
    -- ID values starts at 10001 and auto-increments by 1
    name        varchar(20) not null,
    dept_name   varchar(20),
    salary      numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department (dept_name)
        on delete set null);

insert into instructor
    values ('Smith', 'Comp. Sci.', 80000);

insert into instructor
    values ('Tom', 'Physics', 60000);
```

	ID	name	dept_name	salary
1	10001	Smith	Comp. Sci.	80000.00
2	10002	Tom	Physics	60000.00

Generating Unique Key Values

- MySQL : AUTO_INCREMENT

```
create table instructor(
    ID          int auto_increment, -- MySQL
    name        varchar(20) not null,
    dept_name   varchar(20),
    salary      numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department (dept_name)
        on delete set null);

insert into instructor
    values ('Smith', 'Comp. Sci.', 80000);

insert into instructor
    values ('Tom', 'Physics', 60000);
```

ID	name	dept_name	salary
1	Smith	Comp. Sci.	80,000.00
2	Tom	Physics	60,000.00

Generating Unique Key Values

- In PostgreSQL: use SERIAL or BIGSERIAL
  - **SERIAL**: auto-incremented integer column that takes 4 bytes
  - **BIGSERIAL**: auto-incremented integer column that takes 8 bytes

```
create table instructor(
    ID          serial, -- PostgreSQL
    name        varchar(20) not null,
    dept_name   varchar(20),
    salary      numeric(8, 2) check (salary > 29000),
    primary key (ID),
    foreign key (dept_name) references department (dept_name)
        on delete set null);

insert into instructor
    values ('Smith', 'Comp. Sci.', 80000);

insert into instructor
    values ('Tom', 'Physics', 60000);
```

id [PK] integer	name character varying (20)	dept_name character varying (20)	salary numeric (8,2)
1	Smith	Biology	80000.00
2	Tom	Physics	60000.00

Acknowledgements

- Date and Time
  - Microsoft
    - https://docs.microsoft.com/en-us/sql/t-sql/functions/date-and-time-data-types-and-functions-transact-sql?view=sql-server-ver15
  - MySQL
    - https://dev.mysql.com/doc/refman/8.0/en/date-and-time-types.html
  - PostgreSQL
    - https://www.postgresql.org/docs/current/datatype-datetime.html