CSC 355 Database Systems
Lecture 3

Eric J. Schwabe

School of Computing, DePaul University

Spring 2020

Topics:

- Quick review of Relational Model
- SQL Data Definition Language (DDL)

The Relational Model

- Relations, tuples, attributes, domains
- Relation schema vs. relation instance
- Candidate key, primary key, foreign key
- Database schema vs. database instance
- Domain constraints, key constraints, entity integrity, referential integrity

Writing an SQL Script

- Create file *scriptname*.sql in text editor
- End every SQL statement with a semicolon
- Use SELECT * FROM *TABLENAME* ; statement to display entire contents of a table
- To add comments:
  - -- to begin a one-line comment
  - /* ... */ to begin and end a multi-line comment

Creating a Table

- CREATE TABLE *TABLENAME*
  (*Attribute1 DOMAIN1,*
  Attribute2 DOMAIN2,
  ...
  Attributek DOMAINk);
- Each attribute-domain pair is followed by a comma
- Domain constraints will be enforced

Oracle SQL Domains

- Numerical domains
- String domains
- Dates

Numerical Domains

- General numbers: NUMBER(*x*,*y*)
  - A fixed-precision number with *x* total digits, and *y* digits to the right of the decimal point
    - 101 is NUMBER(3,0)
    - 999.99 is NUMBER(5,2), not NUMBER(3,2)!
- Can also use NUMERIC(*x*,*y*) or DECIMAL(*x*,*y*)
  - Oracle will convert internally to NUMBER(*x*,*y*)

Numerical Domains

- Whole numbers: INTEGER or INT
  - Oracle converts to NUMBER(38,0)
  - To limit size, use NUMBER(*x*,0) or NUMBER(*x*)
- Floating point numbers: FLOAT
  - Can use REAL, Oracle will convert to FLOAT
  - To limit precision, use NUMBER(*x*,*y*)

String Domains

- Fixed-length strings:
  - CHAR(*n*): A fixed-length string of *n* characters
  - Use when you know exact length of strings
- Variable-length strings:
  - VARCHAR(*m*) or VARCHAR2(*m*): A variable-length string of up to *m* characters
  - Oracle will convert internally to VARCHAR2(*m*)
  - Use when you know maximum length of strings

Dates

- DATE:
  - Value given by keyword DATE followed a string in ‘yyyy-mm-dd’ form
    - yyyy = year, mm = month (number), dd = day
    - Always use this general format in your scripts, but some SQL versions may accept other formats too
  - Oracle will convert a string in ‘dd-mon-yyyy’ form to a DATE object
    - dd = day, mon = month (name), yyyy = year

Dropping a Table

- To drop a table:
  DROP TABLE *TABLENAME* ;

- Cannot drop a table if there is a foreign key in another table that references its primary key
  - (…unless you add CASCADE CONSTRAINTS to the command, which will drop the table and remove any constraints that reference it…)

Populating a Table

- To insert a record into a table:

  INSERT INTO *TABLENAME*
  VALUES (*value1, value2, value3, ...*);
- Values of attributes must be given in the same order as in the schema
- Will generate an error if any constraints are violated (domain constraints, key constraints, entity integrity, referential integrity)

Populating a Table

- To insert a record that specifies only some of the attributes:

  INSERT INTO *TABLENAME* (*Attr1, Attr2,...*)
  VALUES (*value1, value2, ...*);

- Missing attributes will be filled in with NULL (unless default values are specified…)

Defaults and Attribute Constraints

- After attribute and domain, before comma:
  - Add default value for the attribute with DEFAULT *value*
  - Disallow NULL values with NOT NULL
  - Impose other constraints with CHECK (*condition*)
    - e.g., to require that attribute is within a range, use CHECK (*value1 <= Attribute AND Attribute <= value2*)
    - Verified whenever a tuple is added or changed

Defining Keys

- For candidate keys, primary keys, and foreign keys:
  - Can add to a single attribute after its domain
  - Can add as a separate CONSTRAINT clause in the CREATE statement when multiple attributes are involved
    - Like attributes, must be followed by commas
    - Constraint can be named by placing CONSTRAINT *Name* in front of it

Defining Candidate Keys

- Use UNIQUE keyword
- Within a single attribute:
  Attribute DOMAIN UNIQUE
- As a separate constraint:
  UNIQUE (*Attribute1*, *Attribute2*, …)
- Key constraints will be enforced

Defining Primary Keys

- Use PRIMARY KEY keywords
- Within a single attribute:
  Attribute DOMAIN PRIMARY KEY
- As a separate constraint:
  PRIMARY KEY (*Attribute1*, *Attribute2*, ...)
- Key constraints and entity integrity will be enforced

Defining Foreign Keys

- Use FOREIGN KEY keywords
- Within a single attribute:

  Attribute DOMAIN
  REFERENCES *TABLE* (*Attribute*)
- As a separate constraint:

  FOREIGN KEY (*Attr1*, *Attr2*, …)
  REFERENCES *TABLE* (*Attr1*, *Attr2*, …)
- Referential integrity will be enforced

Modifying a Schema

- To add or remove attributes and/or constraints:
  ALTER TABLE *TABLENAME*
    - …ADD *Attribute* DOMAIN;
    - …DROP COLUMN *Attribute*;
    - …ADD CONSTRAINT *Name CONSTRAINT* ...;
    - …DROP CONSTRAINT *Name*;

- Constraints must have names to be dropped

Updating Rows

- To modify existing rows in a table:

  UPDATE *TABLENAME*
  SET *Attribute = expression*
  WHERE *condition;*

- Sets *Attribute* to *expression* in exactly those rows that satisfy *condition*

Removing Rows

- To remove existing rows from a table:

  DELETE FROM *TABLENAME*
  WHERE *condition*;

- Removes from the table exactly those rows that satisfy *condition*

Displaying Table Contents

SELECT * FROM *TABLENAME* ;

- This statement will display the entire contents of *TABLENAME* (all rows and columns)
- This is an example of a very simple *query*
- Adding a WHERE clause would let us display only a subset of the rows…

Next:

- Basic SQL Queries