CPSC 5021: Database Systems

Structured Query Language (SQL)

Lin Li

Categories

- Data definition commands

  - create database objects such as tables, indexes

  - define access rights to database objects

- Data manipulation commands

  - insert, update, delete and retrieve data within the database tables

- American National Standards Institute (ANSI) prescribes a standard SQL

- Several SQL dialects exist, but differences are minor

We use MySQL in this course.

Creating a Database

- Need the CREATE privilege
- Syntax:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
```

- Example: create a database named “store”

```
CREATE DATABASE `store`;
```

Creating Table Structures

- Syntax:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
        (create_definition,...)
```

- See http://dev.mysql.com/doc/refman/5.7/en/create-table.html for details

Data Types

- Numeric Type: TINYINT, BOOL, SMALLINT, INT, BIGINT, DECIMAL, FLOAT, DOUBLE, …
- Date and Time Type: DATE, TIMESTAMP, DATETIME, …
  - MySQL retrieves and displays DATE values in 'YYYY-MM-DD' format
  - MySQL retrieves and displays DATETIME values in 'YYYY-MM-DD HH:MM:SS' format
  - The TIMESTAMP data type is used for values that contain both date and time parts.
- String Type: CHAR, VARCHAR(M) (M represents the maximum column length in characters), …
- See http://dev.mysql.com/doc/refman/5.7/en/data-types.html for more details.

Data Types

- Difference between CHAR and VARCHAR

  - The length of a CHAR column is fixed to the length that you declare. The length can be any value from 0 to 255.

  - Values in VARCHAR columns are variable-length strings. The length can be any value from 0 to 65,535

value	char(4)	varchar(4)
‘’	‘ ’	‘’
‘ab’	‘ab ’	‘ab’
‘abcd’	‘abcd’	‘abcd’
‘abcdefgh’	‘abcd’	‘abcd’

Miscellaneous

- NOT NULL: ensures that column does not accept nulls
- UNIQUE: ensures that all values in columns are unique
- PRIMARY KEY
- FOREIGN KEY
- DEFAULT: ensures that attribute uses the default value when its value is not provided in the inserted row
- ……

Example

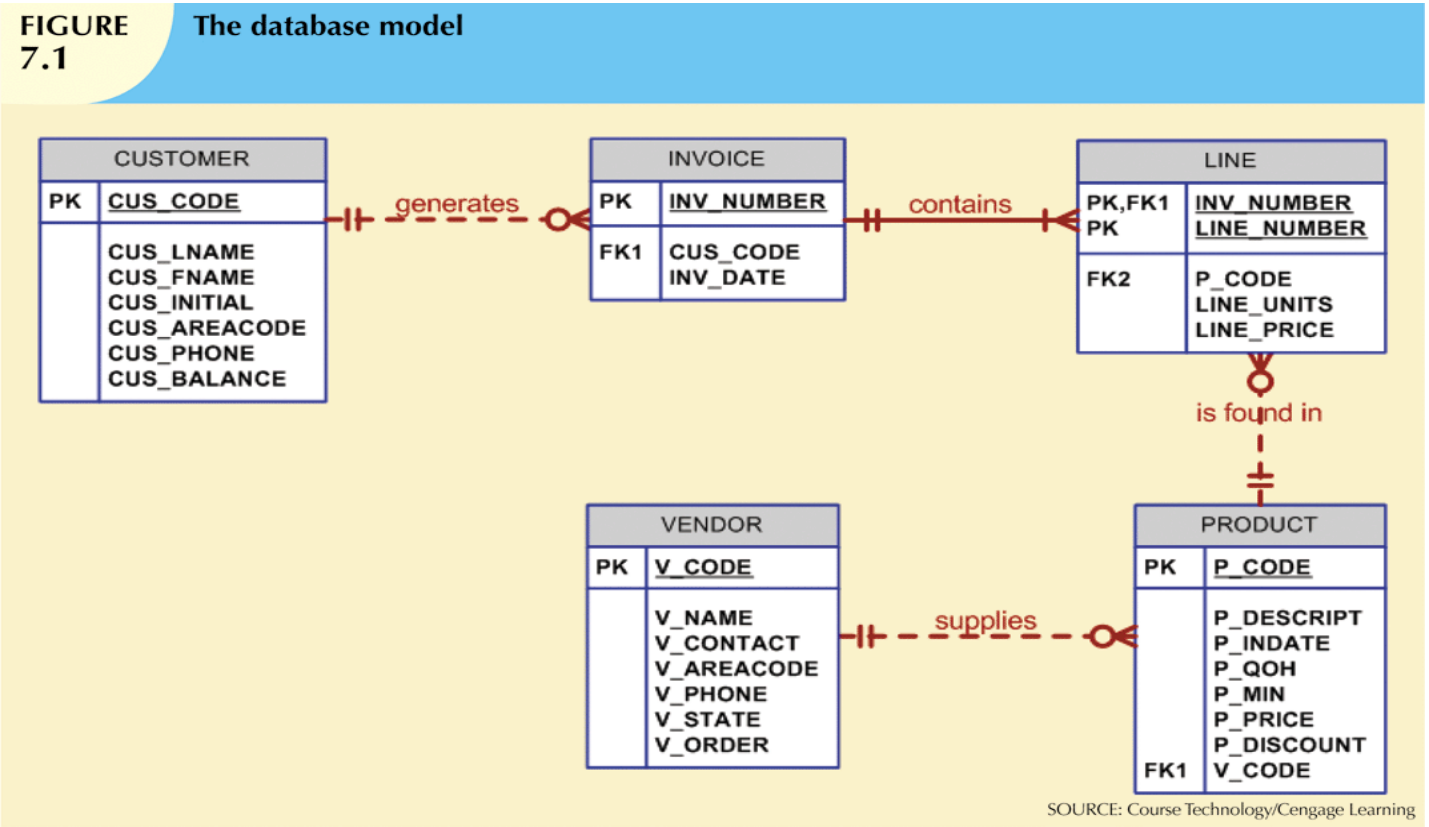
FIGURE 7.1
The database model
CUSTOMER
PK CUS_CODE
CUS_LNAME
CUS_FNAME
CUS_INITIAL
CUS_AREACODE
CUS_PHONE
CUS_BALANCE
generates
INVOICE
PK INV_NUMBER
FK1 CUS_CODE
INV_DATE
contains
LINE
PK,FK1 INV_NUMBER
PK LINE_NUMBER
FK2 P_CODE
LINE_UNITS
LINE_PRICE
is found in
VENDOR
PK V_CODE
V_NAME
V_CONTACT
V_AREACODE
V_PHONE
V_STATE
V_ORDER
supplies
PRODUCT
PK P_CODE
P_DESCRIPT
P_INDATE
P_QOH
P_MIN
P_PRICE
P_DISCOUNT
FK1 V_CODE
SOURCE: Course Technology/Cengage Learning

Creating Table Structures

- Example: create the table VENDOR

```
CREATE TABLE VENDOR(
  V_CODE INT NOT NULL UNIQUE ,
  V_NAME VARCHAR( 35 ) NOT NULL ,
  V_CONTACT VARCHAR( 25 ) NOT NULL ,
  V_AREACODE CHAR( 3 ) NOT NULL ,
  V_PHONE CHAR( 8 ) NOT NULL ,
  V_STATE CHAR( 2 ) NOT NULL ,
  V_ORDER CHAR( 1 ) DEFAULT 'Y' NOT NULL ,
  PRIMARY KEY ( V_CODE )
);
```

Creating Table Structures

- Example: create the table PRODUCT

```
CREATE TABLE PRODUCT(
    P_CODE VARCHAR( 10 ) NOT NULL UNIQUE ,
    P_DESCRIPT VARCHAR( 35 ) NOT NULL ,
    P_INDATE DATE NOT NULL ,
    P_QOH SMALLINT NOT NULL ,
    P_MIN SMALLINT NOT NULL ,
    P_PRICE DECIMAL( 8, 2 ) NOT NULL ,
    P_DISCOUNT DECIMAL( 5, 2 ) NOT NULL ,
    V_CODE INTEGER,
    PRIMARY KEY ( P_CODE ) ,
    FOREIGN KEY ( V_CODE ) REFERENCES VENDOR (V_CODE) ON UPDATE  CASCADE
);
```

Creating Table Structures

- Example: create the table CUSTOMER

```
CREATE TABLE CUSTOMER(
   CUS_CODE     INT PRIMARY KEY ,
   CUS_LNAME    VARCHAR( 15 )    NOT NULL ,
   CUS_FNAME    VARCHAR( 15 )    NOT NULL ,
   CUS_INITIAL  CHAR( 1 ) ,
   CUS_AREACODE   CHAR( 3 )  DEFAULT '615' NOT NULL ,
   CUS_PHONE      CHAR( 8 )  NOT NULL ,
   CUS_BALANCE    DECIMAL( 9, 2 )   DEFAULT 0.00
);
```

Creating Table Structures

- Example: create the table INVOICE

```
CREATE TABLE INVOICE(
   INV_NUMBER INT  PRIMARY KEY ,
   CUS_CODE INT   NOT NULL,
   INV_DATE DATE DEFAULT '2012-01-02' NOT NULL ,
   FOREIGN KEY ( CUS_CODE ) REFERENCES  CUSTOMER(CUS_CODE)
);
```

Creating Table Structures

- Example: create the table LINE

```
CREATE TABLE LINE(
    INV_NUMBER INT NOT NULL ,
    LINE_NUMBER INT NOT NULL ,
    P_CODE VARCHAR( 10 ) NOT NULL ,
    LINE_UNITS DECIMAL( 9, 2 ) DEFAULT 0.00 NOT NULL ,
    LINE_PRICE DECIMAL( 9, 2 ) DEFAULT 0.00 NOT NULL ,
    PRIMARY KEY ( INV_NUMBER, LINE_NUMBER ) ,
    FOREIGN KEY ( INV_NUMBER ) REFERENCES
    INVOICE(INV_NUMBER) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY ( P_CODE ) REFERENCES PRODUCT( P_CODE )
);
```

Exercise – A Consulting Company

- Suppose you are working on a database that stores data for a consulting company that tracks all charges to projects. The charges are based on the hours each employee works on each project. See the handout for the structures and contents of the database.
- Write the SQL code that will create the table structures for the table EMPLOYEE, JOB, ASSIGNMENT, and PROJECT, separately.

Creating Index

- When primary key is declared, DBMS automatically creates unique index.
- Often need additional indexes.
- Indexes can be created on any selected attributes.
- Example:

```
CREATE INDEX P_INDATEX ON PRODUCT(P_INDATE);

CREATE UNIQUE INDEX P_INX ON PRODUCT(P_INDATE, P_CODE);

DROP INDEX P_INX ON PRODUCT;
```

Alter Table

- Change the structure of a table
  - add or delete columns
  - create or destroy indexes
  - change the type of existing columns
  - rename columns or the table itself
  - ……

Alter Table

- Syntax:

```
ALTER TABLE tbl_name [alter_specification [, alter_specification] ...]

alter_specification:
    table_options
| ADD [COLUMN] col_name column_definition [FIRST | AFTER col_name ]
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}
| CHANGE [COLUMN] old_col_name new_col_name column_definition
        [FIRST|AFTER col_name]
| MODIFY [COLUMN] col_name column_definition [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
……
```

- See http://dev.mysql.com/doc/refman/5.7/en/alter-table.html for more details.

Alter Table

- Examples:

```
CREATE TABLE t1 (a INTEGER PRIMARY KEY, b CHAR(10) );

ALTER TABLE t1 RENAME t2;

ALTER TABLE t2 MODIFY a TINYINT NOT NULL, CHANGE b c CHAR(20);

ALTER TABLE t2 ADD d TIMESTAMP;

ALTER TABLE t2 DROP COLUMN c;
```

Alter Table

- Examples:

```
CREATE TABLE PART(
    PART_CODE VARCHAR( 10 ) NOT NULL,
    PART_DESCRIPT VARCHAR( 35 ) NOT NULL ,
    PART_PRICE DECIMAL( 8, 2 ) NOT NULL ,
    V_CODE INTEGER
);

ALTER TABLE PART
ADD PRIMARY KEY ( PART_CODE ) ;

ALTER TABLE PART ADD FOREIGN KEY ( V_CODE ) REFERENCES VENDOR(V_CODE);
```

Drop Table

- Syntax:

```
DROP TABLE tablename
```

- Be careful!!
- Cannot drop a parent table with a foreign key constraint
- Example:

```
DROP TABLE t2
```

Exercise

- Use the SQL on slide 19 to create a table “PART”.
- Then alter the table “PART” using SQL commands to add one attribute “PART_NAME”. You can pick a suitable datatype for this attribute.

```
ALTER TABLE PART ADD PART_NAME VARCHAR( 30 ) ;
```

Data Manipulation Commands

- INSERT
- SELECT
- UPDATE
- DELETE

INSERT

- Enter data into table
- Syntax:

```
INSERT   INTO  tablename   VALUES  (value1, value2, …, valueN);
```

- See http://dev.mysql.com/doc/refman/5.7/en/insert.html for more details.

INSERT

- Examples

Because the PRODUCT table uses its V_CODE to reference the VENDOR table's V_CODE, you need to enter the VENDOR rows before the PRODUCT rows.

```
INSERT INTO VENDOR VALUES ( 21225, 'Bryon, Inc.', 'Smithson', '615', '223-3234', 'TN', 'Y' );
```

-- row contents are entered between parentheses
-- character and date values are entered between apostrophes
-- numerical entries are not enclosed in apostrophes
-- attribute entries are separated by commas
-- a value is required for each column

INSERT

- Examples

```
INSERT INTO VENDOR(V_CODE, V_NAME, V_CONTACT, V_AREACODE, V_PHONE, V_STATE)
VALUES (21228, 'Apple, Inc.', 'Smith', '864', '110-1000', 'SC');
```

```
INSERT INTO VENDOR VALUES (21230, 'Intel, Inc.', 'John', '330', '760-1234', 'CA', '');
```

```
INSERT INTO PRODUCT VALUES ('11AER/31', 'Power Painter, 15 psi, 3-nozzle', '2011-10-12', 8,
5, 109.99, 0.00, 21225);
```

```
INSERT INTO PRODUCT VALUES ('BRT-345', 'Titanium drill bit', '2011-10-18', 75, 10, 4.50,
0.06, NULL);
```

Exercise – A Consulting Company (Cont.)

- Write the SQL code to enter data in each table.

SELECT

- List contents of table
- Syntax:

optional

SELECT columnlist FROM tablename [where conditionlist] ;

-- columnlist represents one or more attributes, separated by commas.

- See http://dev.mysql.com/doc/refman/5.7/en/select.html for more details.

SELECT

- Examples

```
SELECT * FROM PRODUCT;
```

```
SELECT P_CODE, P_DESCRIPT, V_CODE FROM PRODUCT;
```

```
SELECT * FROM VENDOR WHERE V_CODE = 21230;
```

```
SELECT * FROM PRODUCT WHERE P_PRICE > 100;
```

```
SELECT * FROM VENDOR WHERE V_CONTACT = 'Smith'
```

SELECT

- Examples

```
SELECT P_DESCRIPT, P_QOH, P_MIN, P_PRICE  FROM  PRODUCT
WHERE P_INDATE >= '2010-10-12';
```

```
SELECT P_DESCRIPT, P_QOH, P_PRICE, P_QOH * P_PRICE  AS TOTVALUE
FROM PRODUCT;
```

SELECT

- Examples – using logical operators (AND, OR, NOT)

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE V_CODE = 21225 OR V_CODE = 24288
```

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE P_PRICE < 50 AND P_INDATE > '2010-10-01';
```

SELECT

- Examples – using logical operators (AND, OR, NOT)

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE (P_PRICE < 50 AND P_INDATE > '2010-10-01')
OR V_CODE = 21225;
```

```
SELECT * FROM VENDOR
WHERE NOT (V_CODE = 21225);
```

```
SELECT * FROM VENDOR
WHERE V_CODE  != 21225;
```

Exercise – A Consulting Company (Cont.)

- Write the SQL code that will list all attributes in the EMPLOYEE table for a job code of 502.
- Write the SQL code that will list values of “PROJ_NUM””PROJ_NAME” of the PROJECT table.
- Write the SQL code that will list all attributes in the ASSIGNMENT table with ASSIGN_HOURS > 3.
- Write the SQL code that will list ASSIGN_NUM, ASSIGN_DATE, ASSIGN_CHG_HR*ASSIGN_HOURS (*note: name this product as assign_charge*) in the table ASSIGNMENT with ASSIGN_DATE later than 2012-03-21.
- Write the SQL code that will list all attributes in the table “PROJECT” with PROJ_NUM = 22 or PROJ_NUM = 25.

SELECT

- Examples – using special operators (BETWEEN, LIKE, IS NULL, IN, EXISTS)

```
SELECT * FROM PRODUCT
WHERE P_PRICE
BETWEEN 100 AND 200;
```

```
SELECT P_CODE, P_DESCRIPT, V_CODE
FROM PRODUCT
WHERE V_CODE IS NULL;
```

```
SELECT *
FROM PRODUCT
WHERE P_CODE LIKE '1%';
```

"%" is used to match an arbitrary number of characters (including zero characters)

SELECT

- Examples - using special operators (BETWEEN, LIKE, IS NULL, IN, EXISTS)

```
SELECT *
FROM VENDOR
WHERE V_CONTACT LIKE '%N';
```

case-insensitive

```
SELECT *
FROM PRODUCT
WHERE P_CODE LIKE '11_ER/31';
```

"_" is used to match any single character

```
SELECT *
FROM PRODUCT
WHERE P_CODE  NOT LIKE '1%' ;
```

SELECT

- Examples - using special operators (BETWEEN, LIKE, IS NULL, IN, EXISTS)

```
SELECT  *  FROM PRODUCT
WHERE V_CODE IN (21225, 24288);
```

```
SELECT V_CODE, V_NAME
FROM VENDOR
WHERE V_CODE IN (SELECT V_CODE FROM PRODUCT);
```

```
SELECT V_CODE, V_NAME
FROM VENDOR
WHERE EXISTS (SELECT * FROM PRODUCT
WHERE PRODUCT.V_CODE = VENDOR.V_CODE);
```

SELECT

- Examples – ordering a list

```
SELECT P_CODE, P_DESCRIPT, P_INDATE, P_PRICE
FROM PRODUCT
ORDER BY P_PRICE;
```

ASC by default

```
SELECT P_CODE, P_DESCRIPT, P_INDATE, P_PRICE
FROM PRODUCT
ORDER BY P_PRICE DESC;
```

Exercise – A Consulting Company (Cont.)

- Write the SQL code required to list all employees whose last names start with Smith.
- Write the SQL code to select from the table “JOB” all jobs which job_code value appear in the EMPLOYEE table.
- Write the SQL code that will produce a listing for the data in the EMPLOYEE table in descending order by EMP_YEAR.