



---

# **CS 309A- Database Management Systems**



# Introduction to SQL

---

- ◇ Structured query language (SQL)
  - Standard query language for relational databases
    - American National Standards Institute (ANSI) prescribes a standard SQL
    - Several SQL dialects exist
  - Enables users to create database objects and manipulate and view data
  - Basic command set has vocabulary of fewer than 100 words
- ◇ Basic categories for SQL commands
  - Data definition language (DDL)
  - Data manipulation language (DML)
- ◇ Reserved words
  - SQL command words

# Data Definition Commands

**TABLE  
7.1**

**SQL Data Definition Commands**

COMMAND OR OPTION	DESCRIPTION
CREATE SCHEMA AUTHORIZATION	Creates a database schema
CREATE TABLE	Creates a new table in the user's database schema
NOT NULL	Ensures that a column will not have null values
UNIQUE	Ensures that a column will not have duplicate values
PRIMARY KEY	Defines a primary key for a table
FOREIGN KEY	Defines a foreign key for a table
DEFAULT	Defines a default value for a column (when no value is given)
CHECK	Validates data in an attribute
CREATE INDEX	Creates an index for a table
CREATE VIEW	Creates a dynamic subset of rows and columns from one or more tables (see Chapter 8, Advanced SQL)
ALTER TABLE	Modifies a table's definition (adds, modifies, or deletes attributes or constraints)
CREATE TABLE AS	Creates a new table based on a query in the user's database schema
DROP TABLE	Permanently deletes a table (and its data)
DROP INDEX	Permanently deletes an index
DROP VIEW	Permanently deletes a view

# Data Manipulation Commands

**TABLE 7.2** SQL Data Manipulation Commands

COMMAND OR OPTION	DESCRIPTION
INSERT	Inserts row(s) into a table
SELECT	Selects attributes from rows in one or more tables or views
WHERE	Restricts the selection of rows based on a conditional expression
GROUP BY	Groups the selected rows based on one or more attributes
HAVING	Restricts the selection of grouped rows based on a condition
ORDER BY	Orders the selected rows based on one or more attributes
UPDATE	Modifies an attribute's values in one or more table's rows
DELETE	Deletes one or more rows from a table
COMMIT	Permanently saves data changes
ROLLBACK	Restores data to their original values
<b>Comparison operators</b>	
=, <, >, <=, >=, <>	Used in conditional expressions
<b>Logical operators</b>	
AND/OR/NOT	Used in conditional expressions
<b>Special operators</b>	Used in conditional expressions
BETWEEN	Checks whether an attribute value is within a range
IS NULL	Checks whether an attribute value is null
LIKE	Checks whether an attribute value matches a given string pattern
IN	Checks whether an attribute value matches any value within a value list
EXISTS	Checks whether a subquery returns any rows
DISTINCT	Limits values to unique values
<b>Aggregate functions</b>	Used with SELECT to return mathematical summaries on columns
COUNT	Returns the number of rows with non-null values for a given column
MIN	Returns the minimum attribute value found in a given column
MAX	Returns the maximum attribute value found in a given column
SUM	Returns the sum of all values for a given column
AVG	Returns the average of all values for a given column



# Data Types

---

## ◇ Data type

- Specifies kind of data that column stores
- Provides means for error checking
- Enable DBMS to use storage space more efficiently
- Basic types
  - String Types
  - Numeric Types
  - Date and Time Types

# String Types

## ◇ VARCHAR

- Variable-length character data
- *columnname* VARCHAR(*maximum\_size*)

## ◇ CHAR

- Fixed-length character data
- *columnname* CHAR(*maximum\_size*)

VARCHAR values are stored as a 1-byte or 2-byte length prefix plus data.

Value	CHAR ( 4 )	Storage Required	VARCHAR ( 4 )	Storage Required
' '	' '	4 bytes	' '	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

More detail: <https://dev.mysql.com/doc/refman/5.7/en/char.html>



# Numeric Types

## ◇ Integer Types (Exact Value)

*columnname* INT

Type	Storage	Minimum Value	Maximum Value
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615



# Numeric Types

---

## ◇ Fixed-Point Type (Exact Value)

- DECIMAL/NUMERIC

*columnname* DECIMAL (*precision*, *scale*)

*columnname* NUMERIC (*precision*, *scale*)

- *Precision*--the number of significant digits that are stored for values
- *Scale*--the number of digits that can be stored following the decimal point
- For example, the number 123.45 has a **precision** of 5 and a **scale** of 2.
- These types are used when it is important to preserve exact precision, for example with monetary data.





# Numeric Types

---

## ◇ Floating-Point Types (Approximate Value)

- FLOAT (4 bytes)
- DOUBLE PRECISION (8 bytes)

*columnname* FLOAT (*precision*, *scale*)

*columnname* DOUBLE PRECISION (*precision*,  
*scale*)



# Date and Time Data Types

---

## ◇ DATE

- Used for values with a date part but no time part
- 'YYYY-MM-DD'

## ◇ DATETIME

- Used for values that contain both date and time parts
- 'YYYY-MM-DD HH:MM:SS'

## ◇ TIMESTAMP

- Used for values that contain both date and time parts
- Automatic initialization and updating to the current date and time

## ◇ TIME

- Used for time value
- 'HHH:MM:SS'
- The hours part may be so large because it can be used to represent elapsed time or a time interval between two events.

## ◇ YEAR

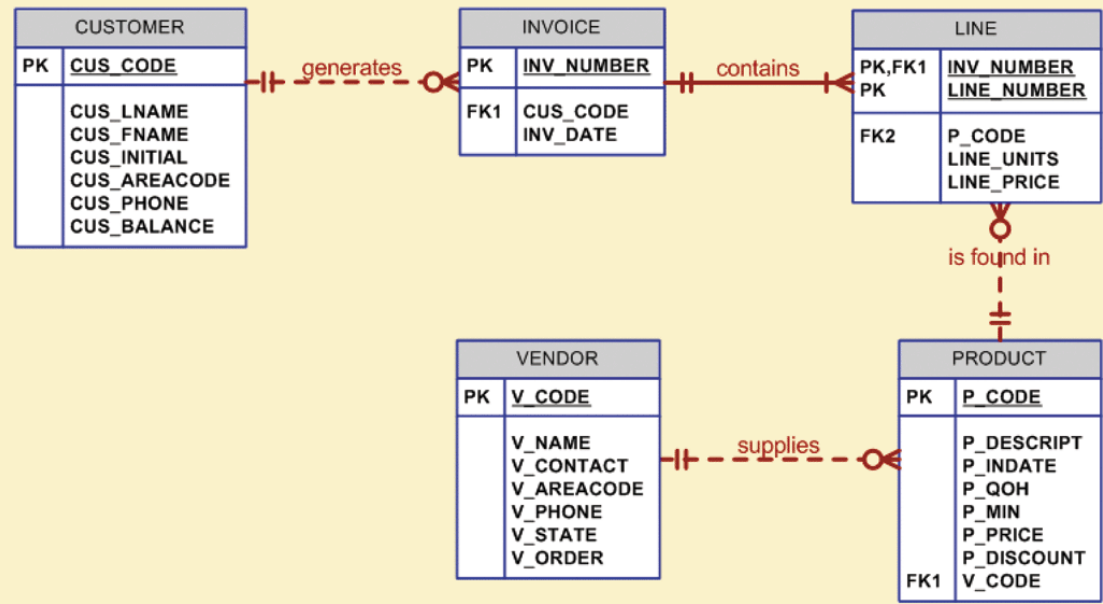
- Used for year value
- 'YYYY'

# Database model

◇ We will use a simple database (sale) with the following tables to illustrate commands:

- CUSTOMER
- INVOICE
- LINE
- PRODUCT
- VENDOR

FIGURE 7.1 The database model



SOURCE: Course Technology/Cengage Learning



# Connecting and Disconnecting to the MySQL

---

## ◇ Authentication

- DBMS verifies that only registered users are able to access database
- Log on to RDBMS using user ID and password created by database administrator

## ◇ To connect to the MySQL server:

```
mysql -h hostname -u username -p
```

- *hostname*: where the MySQL server is running
- *username*: MySQL account
- -h: specifies that you are giving the host name of MySQL server
- -u: specifies that you are giving a login
- -p: specifies that you login requires a password

## ◇ To disconnect MySQL: quit



# Basic MySQL command

---

- ◇ MySQL command is **case insensitive**
- ◇ **Most** MySQL commands end with semicolon (;). If you do not end your command with a semicolon (;), MySQL assumes you want to keep typing.

# List all databases

---

◇ SHOW DATABASES;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```



# Create a database

---

- ◇ Two tasks must be completed:
  - Create database structure – data dictionary
  - Create tables that will hold end-user data
- ◇ Schema
  - A logical group of database objects that are related to each other
  - the schema belongs to a single user or application

# Create a database

---

◇ CREATE DATABASE *database\_name*;

```
mysql> create database sale;  
Query OK, 1 row affected (0.04 sec)
```

```
mysql> show databases;
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sale |  
| sys |  
+-----+
```

```
5 rows in set (0.00 sec)
```





## Select a database

---

◇ `USE database_name`

```
mysql> use sale;  
Database changed
```

## Delete a database

---

◇ DROP DATABASE *database\_name*;

```
mysql> drop database sale;  
Query OK, 0 rows affected (0.13 sec)
```

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sys |  
+-----+  
4 rows in set (0.00 sec)
```



# Tables

---

- ◇ List all tables in the selected database:

`SHOW TABLES;`

```
mysql> show tables;  
Empty set (0.01 sec)
```

- ◇ Create a table

```
CREATE TABLE table_name  
(column_name1 data_type,  
Column_name2 data_type, ...);
```



# Examples

```
mysql> create table CUSTOMER (  
    -> CUS_CODE int,  
    -> CUS_LNAME varchar(15),  
    -> CUS_FNAME varchar(15),  
    -> CUS_INITIAL varchar(1),  
    -> CUS_AREACODE varchar(3),  
    -> CUS_PHONE varchar(8),  
    -> CUS_BALANCE decimal(8,2)  
    -> );
```

Query OK, 0 rows affected (0.14 sec)

```
mysql> show tables;  
+-----+  
| Tables_in_sale |  
+-----+  
| customer       |  
+-----+  
1 row in set (0.00 sec)
```



# View information about the table

◇ DESCRIBE *tablename*;

```
mysql> describe customer;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CUS_CODE       | int(11)       | YES  |     | NULL    |       |
| CUS_LNAME      | varchar(15)   | YES  |     | NULL    |       |
| CUS_FNAME      | varchar(15)   | YES  |     | NULL    |       |
| CUS_INITIAL    | varchar(1)    | YES  |     | NULL    |       |
| CUS_AREACODE   | varchar(3)    | YES  |     | NULL    |       |
| CUS_PHONE      | varchar(8)    | YES  |     | NULL    |       |
| CUS_BALANCE    | decimal(8,2)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)
```



# Delete and Rename an existing table

---

## ◇ Delete an existing table

- `DROP TABLE tablename;`

```
mysql> drop table customer;  
Query OK, 0 rows affected (0.06 sec)
```

## • Rename an existing table

- `RENAME TABLE old_tablename TO new_tablename;`

# Continue creating tables: VENDOR & PRODUCT

**FIGURE 7.2** The VENDOR and PRODUCT tables

Table name: VENDOR

Database name: Ch07\_SaleCo

V_CODE	V_NAME	V_CONTACT	V_AREACODE	V_PHONE	V_STATE	V_ORDER
21225	Bryson, Inc.	Smithson	615	223-3234	TN	Y
21226	SuperLoo, Inc.	Flushing	904	215-8995	FL	N
21231	D&E Supply	Singh	615	228-3245	TN	Y
21344	Gomez Bros.	Ortega	615	889-2546	KY	N
22567	Dome Supply	Smith	901	678-1419	GA	N
23119	Randsets Ltd.	Anderson	901	678-3998	GA	Y
24004	Brackman Bros.	Browning	615	228-1410	TN	N
24288	ORDVA, Inc.	Hakford	615	898-1234	TN	Y
25443	B&K, Inc.	Smith	904	227-0093	FL	N
25501	Damal Supplies	Smythe	615	890-3529	TN	N
25595	Rubicon Systems	Orton	904	456-0092	FL	Y

Table name: PRODUCT

P_CODE	P_DESCRIPTION	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-11	8	5	109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-11	32	15	14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-11	18	12	17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-12	15	8	39.95	0.00	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-12	23	5	43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-11	8	5	109.92	0.05	24288
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-11	6	5	99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-12	12	5	38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-12	23	10	9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-12	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-11	43	20	4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-12	11	5	256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-12	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-12	172	75	6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-12	237	100	8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/8", .5" mesh	17-Jan-12	18	5	119.95	0.10	25595

SOURCE: Course Technology/Cengage Learning

❖ Why VENDOR must be created before PRODUCT?



# Table VENDOR

---

▪ V_CODE	NUMBER? VARCHAR?
▪ V_NAME	VARCHAR(35)
▪ V_CONTACT	VARCHAR(25)
▪ V_AREACODE	NUMBER? CHAR?
▪ V_PHONE	CHAR(8)
▪ V_STATE	CHAR(2)
▪ V_ORDER	CHAR(1)

Recall data types we introduced:

String

- VARCHAR(size)
- CHAR(size)

Numeric

- INT/SMALLINT/BIGINT
- DECIMAL/NUMERIC(dig\_len, deci\_len)
- FLOAT/DOUBLE(dig\_len, deci\_len)

Date/time

- DATE/DATETIME/TIMESTAMP
- TIME
- YEAR





# Choose Character type or Number type?

---

- ◇ General Principle: if you want to do arithmetic operations on this attribute, choose Numeric type. Otherwise, choose String type.
  - V\_CODE
    - If you want the computer to generate new vendor codes by adding 1 to the existing largest vendor code, use INT
    - If you do not want to perform arithmetic operations on V\_CODE, use VARCHAR
  - V\_AREACODE
    - Arithmetic operations on area codes does not yield meaningful results, use CHAR(3)



---

## Table VENDOR

- V\_CODE           INT
- V\_NAME          VARCHAR(35)
- V\_CONTACT      VARCHAR(25)
- V\_AREACODE     CHAR(3)
- V\_PHONE         CHAR(8)
- V\_STATE         CHAR(2)
- V\_ORDER         CHAR(1)

## Table PRODUCT

- P\_CODE            VARCHAR(10)
- P\_DESCRIPT        VARCHAR(35)
- P\_INDATE          DATE
- P\_QOH             INT
- P\_MIN             INT
- P\_PRICE           DECIMAL(8,2)
- P\_DISCOUNT      DECIMAL(5,2)
- V\_CODE            INT



# Constraints

---

- ◇ Rules that restrict data values that can be entered into column.
- ◇ Types of constraints:
  - Integrity constraints
  - Value constraints
- ◇ Another classification:
  - Table constraint
    - Restricts data value with respect to all other values in table
  - Column constraint
    - Limits value that can be placed in specific column
    - Irrespective of values that exist in other table rows
- ◇ Constraint definitions should be placed either:
  - At end of CREATE TABLE command after table columns declared
  - Within each column definition



# Integrity constraints: set Primary Key

## ◇ Primary key

- Table constraint
- Syntax (within column definition)
  - PRIMARY KEY
- Syntax (at the end of table definition)
  - PRIMARY KEY (*columnname1*, [*columnname2*, ...])

```
CREATE TABLE VENDOR (  
    V_CODE int PRIMARY KEY,  
    V_NAME varchar(35),  
    V_CONTACT varchar(25),  
    V_AREACODE char(3),  
    V_PHONE char(8),  
    V_STATE char(2),  
    V_ORDER char(1)  
);
```

```
CREATE TABLE CUSTOMER (  
    CUS_CODE int,  
    CUS_LNAME varchar(15),  
    CUS_FNAME varchar(15),  
    CUS_INITIAL varchar(1),  
    CUS_AREACODE varchar(3),  
    CUS_PHONE varchar(8),  
    CUS_BALANCE decimal(8,2),  
    PRIMARY KEY (CUS_CODE)  
);
```



# Example

```
mysql> create table vendor (  
  -> U_CODE INT PRIMARY KEY,  
  -> U_NAME VARCHAR (35),  
  -> U_CONTACT VARCHAR (25),  
  -> U_AREACODE CHAR (3),  
  -> U_PHONE CHAR (8),  
  -> U_STATE CHAR (2),  
  -> U_ORDER CHAR (1)  
  -> );
```

Query OK, 0 rows affected (0.10 sec)

```
mysql> describe vendor;
```

Field	Type	Null	Key	Default	Extra
U_CODE	int(11)	NO	PRI	NULL	
U_NAME	varchar(35)	YES		NULL	
U_CONTACT	varchar(25)	YES		NULL	
U_AREACODE	char(3)	YES		NULL	
U_PHONE	char(8)	YES		NULL	
U_STATE	char(2)	YES		NULL	
U_ORDER	char(1)	YES		NULL	

7 rows in set (0.00 sec)



# Integrity constraints: set Foreign Keys

---

## ◇ Foreign key

- Column constraint
- Specifies that value user inserts in column must exist as primary key in referenced table
- Syntax (placed at end of table definition)

```
[CONSTRAINT constraint_name]  
FOREIGN KEY (columnname)  
REFERENCES primary_key_tablename  
           (primary_key_columnname)
```



# Constraint name

---

- ◇ Constraint naming convention
  - *tablename\_columnname\_constraintid*
- ◇ constraintid

Constraint Type	ConstraintID Abbreviation
PRIMARY KEY	pk
FOREIGN KEY	fk
CHECK CONDITION	cc
NOT NULL	nn
UNIQUE	uk



# Integrity constraints: set Foreign Keys

---

```
CREATE TABLE PRODUCT (  
    P_CODE varchar(10),  
    P_DISCRIPT varchar(35),  
    P_INDATE date,  
    P_QOH int,  
    P_MIN int,  
    P_PRICE decimal(8,2),  
    P_DISCOUNT decimal(5,2),  
    V_CODE int,  
    PRIMARY KEY (P_CODE),  
    CONSTRAINT PRODUCT_V_CODE_FK  
    FOREIGN KEY (V_CODE)  
    REFERENCES VENDOR (V_CODE)  
);
```

The foreign key constraint definition ensures that:

- You cannot insert a product with a invalid vendor code.
- You cannot delete a vendor from the VENDOR table if at least product row references that vendor.



# Example



```
mysql> create table product (  
-> P_CODE VARCHAR (10),  
-> P_DISCRIPT VARCHAR (35),  
-> P_INDATE DATE,  
-> P_QOH INT,  
-> P_MIN INT,  
-> P_PRICE DECIMAL (8,2),  
-> P_DISCOUNT DECIMAL (5,2),  
-> U_CODE INT,  
-> PRIMARY KEY (P_CODE),  
-> CONSTRAINT PRODUCT_U_CODE_FK  
-> FOREIGN KEY (U_CODE)  
-> REFERENCES UENDOR (U_CODE)  
-> );
```

Query OK, 0 rows affected (0.12 sec)

```
mysql> show tables;
```

```
+-----+  
! Tables_in_sale !  
+-----+  
! product        !  
! vendor         !  
+-----+
```

2 rows in set (0.00 sec)



# Value constraints

---

- ◇ Column-level constraints
- ◇ Restrict data values that users can enter
- ◇ Commonly used value constraints

- CHECK conditions

it checks to see a specified condition exists

i.e., the minimum order value must be at least \$75.00 for free shipping

- NOT NULL constraint

a column does not accept nulls

- DEFAULT constraint

assigns a value to an attribute when a new row is added to a table

- UNIQUE constraint

all values in a column are unique



# Table CUSTOMER

---

```
mysql> create table customer (  
    -> CUS_CODE INT PRIMARY KEY,  
    -> CUS_LNAME VARCHAR (15) NOT NULL,  
    -> CUS_FNAME VARCHAR (15) NOT NULL,  
    -> CUS_INITIAL VARCHAR (1),  
    -> CUS_AREACODE VARCHAR (3) DEFAULT '615' NOT NULL,  
    -> CUS_PHONE VARCHAR (8) NOT NULL,  
    -> CUS_BALANCE DECIMAL (8,2) DEFAULT 0.00,  
    -> CHECK (CUS_AREACODE IN ('615', '713', '931')),  
    -> UNIQUE (CUS_LNAME, CUS_FNAME)  
    -> );
```

```
Query OK, 0 rows affected (0.09 sec)
```



# Table CUSTOMER

```
mysql> describe customer;
```

Field	Type	Null	Key	Default	Extra
CUS_CODE	int(11)	NO	PRI	NULL	
CUS_LNAME	varchar(15)	NO	MUL	NULL	
CUS_FNAME	varchar(15)	NO		NULL	
CUS_INITIAL	varchar(1)	YES		NULL	
CUS_AREACODE	varchar(3)	NO		615	
CUS_PHONE	varchar(8)	NO		NULL	
CUS_BALANCE	decimal(8,2)	YES		0.00	

```
7 rows in set (0.00 sec)
```



# Table INVOICE

```
mysql> create table invoice (<br>  -> INU_NUMBER INT PRIMARY KEY,<br>  -> CUS_CODE INT NOT NULL,<br>  -> INU_DATE DATETIME DEFAULT CURRENT_TIMESTAMP NOT NULL,<br>  -> CONSTRAINT INVOICE_CUS_CODE_FK<br>  -> FOREIGN KEY (CUS_CODE)<br>  -> REFERENCES CUSTOMER (CUS_CODE),<br>  -> CONSTRAINT INVOICE_INU_DATE_CC CHECK<br>  -> (INU_DATE > TO_DATE ('2012-01-01', 'YYYY-MM-DD'))<br>  -> );
```

Query OK, 0 rows affected (0.13 sec)

```
mysql> describe invoice;
```

Field	Type	Null	Key	Default	Extra
INU_NUMBER	int(11)	NO	PRI	NULL	
CUS_CODE	int(11)	NO	MUL	NULL	
INU_DATE	datetime	NO		CURRENT_TIMESTAMP	

3 rows in set (0.00 sec)



## Table INVOICE

---

- ◇ You can give a name to a value constraint:

```
CONSTRAINT INVOICE_INV_DATE_CC CHECK  
(INV_DATE > TO_DATE('2012-01-01', 'YYYY-MM-DD'))
```

- ◇ If you want to set the default value for date type column as system time:
  - You cannot set the default for a DATE column to be the value of a function such as NOW() or CURRENT\_DATE.
  - You can specify CURRENT\_TIMESTAMP as the default for TIMESTAMP and DATETIME columns.

# Table LINE

```
mysql> create table LINE (  
  -> INU_NUMBER INT NOT NULL,  
  -> LINE_NUMBER SMALLINT 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 (INU_NUMBER, LINE_NUMBER),  
  -> CONSTRAINT LINE_INU_NUMBER_FK FOREIGN KEY (INU_NUMBER)  
  -> REFERENCES INVOICE (INU_NUMBER) ON DELETE CASCADE,  
  -> CONSTRAINT LINE_P_CODE_FK FOREIGN KEY (P_CODE)  
  -> REFERENCES PRODUCT (P_CODE)  
  -> );  
Query OK, 0 rows affected (0.10 sec)
```

- The ON DELETE CASCADE is recommended for weak entities to ensure that the deletion of a row in the strong entity automatically triggers the deletion of the corresponding rows in the dependent weak entity.

# Thank you & Questions

