



Dalhousie University
Faculty of Computer Science

CSCI 2141 – Intro to Database Systems

Week 4 – Introduction to
Structured Query Language (SQL)

Introduction to SQL

- SQL stands for **Structured Query Language**
 - Pronounced as *S-Q-L* or *Sequel*
- Composed of commands that enable users to:
 - Create database and table structures
 - Perform data manipulation and data administration
 - Query the database to extract useful information
- All relational DBMS support SQL
 - Vendors have developed extensions to the basic SQL command set

Introduction to SQL

- Relatively easy to learn
 - Basic command set has a vocabulary of fewer than 100 words
 - SQL is a non-procedural language
 - You command *what* is to be done, not *how* it is to be done
 - Example: A single command creates the complex table structures required to store and manipulate data
 - End-users do not need to know the physical data storage format or the complex activities that take place when a SQL command is executed
-

Introduction to SQL

- Categories of SQL functions:
 - Data definition language (DDL)
 - Data manipulation language (DML)
- Nonprocedural language with basic command vocabulary set of less than 100 words
- Differences in SQL dialects are minor

SQL – Data Definition Language (DDL)

- DDL includes commands to create database objects such as:
 - Tables
 - Indexes
 - Views
 - Also includes commands to define access rights to the database objects
 - DDL is used to create and manage the structure of the database
-

SQL Data Definition Command

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

SQL – Data Manipulation Language (DML)

- DML is used to query and manage the data inside the database
 - DML includes commands to insert, update and delete data within the database table
 - DML also includes commands to query the database and return useful information
-

SQL 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 its original values
Comparison operators	
=, <, >, <=, >=, <>, !=	Used in conditional expressions
Logical operators	
AND/OR/NOT	Used in conditional expressions
Special operators	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
Aggregate functions	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

SQL Standardization

- ANSI prescribes a standard SQL
 - ANSI SQL accepted by ISO
 - Adherence to ANSI/ISO SQL required in commercial and government contract database specifications
 - Many vendors add their own enhancements
 - Seldom possible to move a SQL-based application from one RDBMS to another without making changes
 - Several different dialects of SQL
 - Differences are minor
 - Easy to adapt to a dialect once you know the basics
-

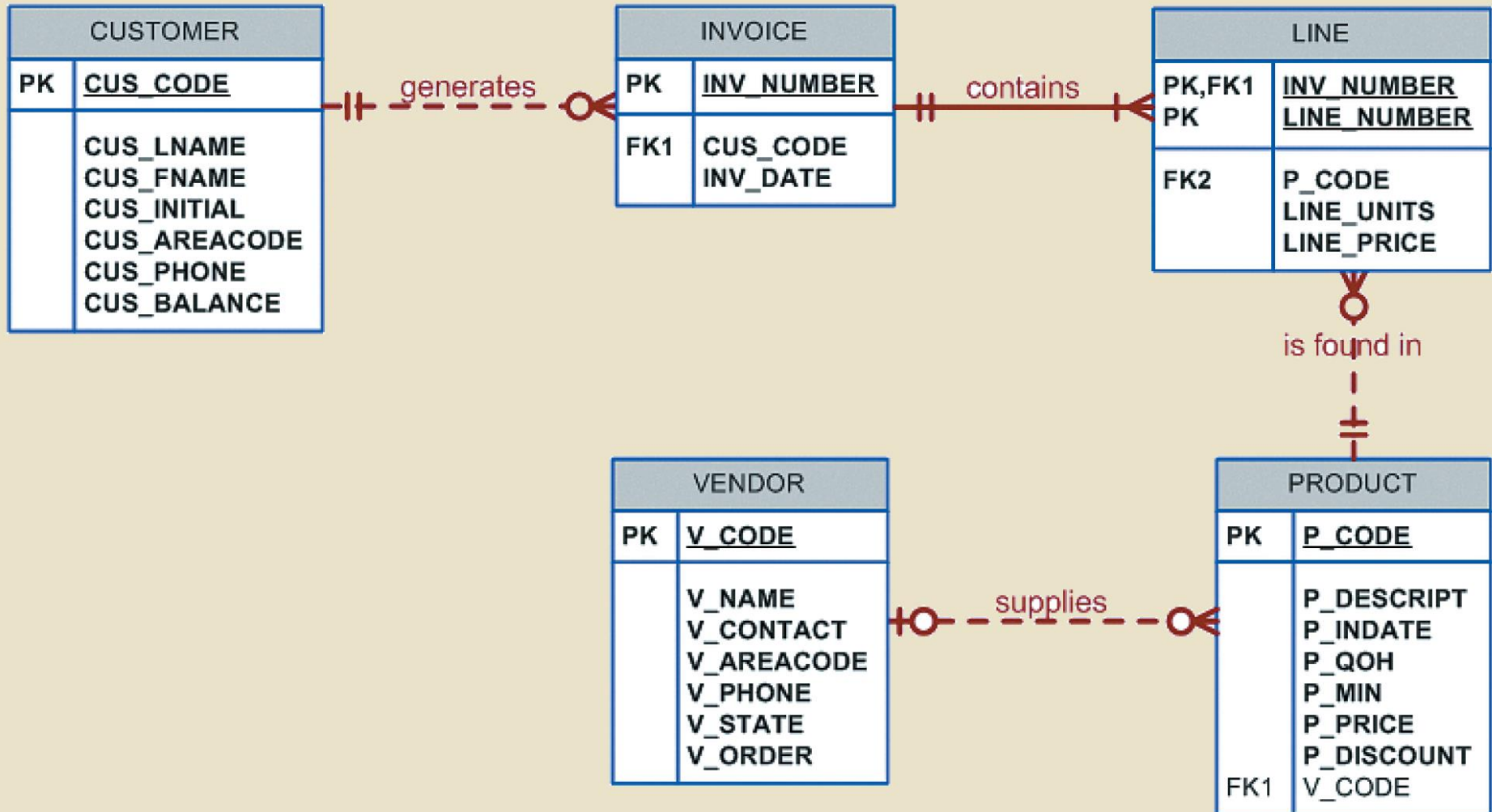
Sample Database

for in-class use

The Database Model – Business Rules

- A customer may generate many invoices. Each invoice is generated by one customer
 - An invoice contains one or more invoice lines. Each invoice line is associated with one invoice
 - Each invoice line references one product. A product may be found in many invoice lines
 - A vendor *may* supply many products. Some vendors do not yet supply products.
 - If a product is vendor-supplied, it is supplied by only a single vendor
 - Some products are not supplied by a vendor
-

The Database Model



Sample 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

Table name: PRODUCT

P_CODE	P_DESCRIPT	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-15	32	15	14.99	0.05	21344
23114-AA	Sledge hammer, 12 lb.	02-Jan-16	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-15	43	20	4.99	0.00	21344
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-16	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-16	172	75	6.99	0.00	21225
SW-23116	2.5-in. wdl. screw, 50	24-Feb-16	237	100	8.45	0.00	21231

SQL's Data Definition Language (DDL)

Creating the Database

- Create database structure
 - RDBMS creates physical files that will hold database
 - Differs from one RDBMS to another
- **Authentication** is the process DBMS uses to verify that only registered users access the database
 - Required for the creation of tables
 - User should log on to RDBMS using user ID and password created by database administrator

Creating the Database

- Creating the physical files that hold the database means interacting with the OS and the file system of that OS
 - This process can vary substantially from vendor to vendor
 - Easy to create in Access
 - Enterprise RDBMS have greater security requirements and complexity, hence creating a database is a more elaborate process
 - With the exception of creating the database, most SQL commands used by all vendors deviate little from ANSI/ISO SQL
-

The Database Schema

- Logical group of database objects – such as tables and indexes - related to each other
- Command:
 - `CREATE SCHEMA schema_name AUTHORIZATION {creator};`
 - `CREATE SCHEMA my_first_db AUTHORIZATION khurram;`
 - DATABASE can be used in place of SCHEMA
 - Optionally, can use [IF NOT EXISTS] to prevent error message if database already exists
 - `CREATE DATABASE IF NOT EXISTS my_first_db;`

Creating Table Structures

- Syntax

CREATE TABLE *tablename* (

column1 *data type* [*constraint*],

column2 *data type* [*constraint*],

PRIMARY KEY (*column1* [, *column2*]),

FOREIGN KEY (*column1* [, *column2*])

REFERENCES *tablename* (*column1* [, *column2*]),

CONSTRAINT *constraint1*);

Creating the VENDOR Table

```
CREATE TABLE VENDOR (  
    V_CODE          INT          NOT NULL    UNIQUE,  
    V_NAME          VARCHAR(15) NOT NULL,  
    V_CONTACT       VARCHAR(50) NOT NULL,  
    V_AREACODE      CHAR(3)      NOT NULL,  
    V_PHONE         CHAR(8)      NOT NULL,  
    V_STATE         CHAR(2)      NOT NULL,  
    V_ORDER         CHAR(1)      NOT NULL,  
    PRIMARY KEY     (V_CODE)  
);
```

Creating the PRODUCT Table

```
CREATE TABLE PRODUCT (
```

```
    P_CODE          VARCHAR(10)    NOT NULL        UNIQUE,
```

```
    P_DESCRIPT      VARCHAR(35)    NOT NULL,
```

```
    P_INDATE        DATETIME       NOT NULL,
```

```
    P_QOH           SMALLINT       NOT NULL,
```

```
    P_MIN           SMALLINT       NOT NULL,
```

```
    P_PRICE         NUMERIC(8,2)   NOT NULL,
```

```
    P_DISCOUNT     NUMERIC(4,2)   NOT NULL,
```

```
    V_CODE          INTEGER,
```

```
    PRIMARY KEY     (P_CODE),
```

```
    FOREIGN KEY     (V_CODE)        REFERENCES    VENDOR (V_CODE)
```

```
);
```

SQL Constraints

NOT NULL

- Ensures that column does not accept nulls

UNIQUE

- Ensures that all values in column are unique

DEFAULT

- Assigns value to attribute when a new row is added to table

CHECK

- Validates data when attribute value is entered

Constraints

UNIQUE Constraint:

```
CONSTRAINT CUS_UI1 UNIQUE (CUS_LNAME, CUS_FNAME)
```

CHECK Constraint:

```
CONSTRAINT INV_CK1 CHECK (INV_DATE > TO_DATE ('01-JAN-2016', 'DD-MON-YYYY'))
```

Creating Table Structures – Standard Practices

- Use one line per column (attribute) definition
- Use spaces to line up attribute characteristics and constraints
- Table and attribute names are fully capitalized
- Table definition enclosed in parentheses
- RDBMS automatically enforces referential integrity for foreign keys.

Common SQL Data Types

DATA TYPE	FORMAT	COMMENTS
Numeric	NUMBER(L,D) or NUMERIC(L,D)	The declaration NUMBER(7,2) or NUMERIC(7,2) indicates that numbers will be stored with two decimal places and may be up to seven digits long, including the sign and the decimal place (for example, 12.32 or –134.99).
	INTEGER	May be abbreviated as INT. Integers are (whole) counting numbers, so they cannot be used if you want to store numbers that require decimal places.
	SMALLINT	Like INTEGER but limited to integer values up to six digits. If your integer values are relatively small, use SMALLINT instead of INT.
	DECIMAL(L,D)	Like the NUMBER specification, but the storage length is a <i>minimum</i> specification. That is, greater lengths are acceptable, but smaller ones are not. DECIMAL(9,2), DECIMAL(9), and DECIMAL are all acceptable.
Character	CHAR(L)	Fixed-length character data for up to 255 characters. If you store strings that are not as long as the CHAR parameter value, the remaining spaces are left unused. Therefore, if you specify CHAR(25), strings such as <i>Smith</i> and <i>Katzenjammer</i> are each stored as 25 characters. However, a U.S. area code is always three digits long, so CHAR(3) would be appropriate if you wanted to store such codes.
	VARCHAR(L) or VARCHAR2(L)	Variable-length character data. The designation VARCHAR2(25) or VARCHAR(25) will let you store characters up to 25 characters long. However, unlike CHAR, VARCHAR will not leave unused spaces. Oracle automatically converts VARCHAR to VARCHAR2.
Date	DATE	Stores dates in the Julian date format.

SQL Date and DateTime Data Types

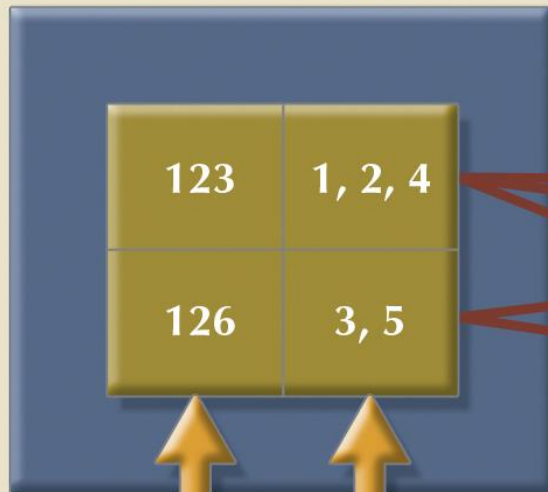
- The DATE type is used for values with a date part but no time part.
 - MySQL retrieves and displays DATE values in 'YYYY-MM-DD' format.
 - The supported range is '1000-01-01' to '9999-12-31'.
 - The DATETIME type is used for values that contain both date and time parts.
 - MySQL retrieves and displays DATETIME values in 'YYYY-MM-DD HH:MM:SS' format.
 - The supported range is '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.
 - TIMESTAMP is used to record the date and time of an event
 - The time zone used is the server's time zone
 - The supported range is '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC
-

Indexes

- Indexes are created to provide quick access to data
 - Orderly arrangement to logically access rows in a table
- **Index key:** Index's reference point that leads to data location identified by the key
- **Unique index:** Index key can have only one pointer value associated with it
- Each index is associated with only one table
 - One table can have several indexes
 - Index is automatically created on the primary key column

Indexes

PAINTING table index



**PAINTER_NUM
(index key)**

**Pointers to the
PAINTING
table rows**

PAINTING table

PAINTING_NUM	PAINTING_TITLE	PAINTER_NUM
1338	Dawn Thunder	123
1339	Vanilla Roses To Nowhere	123
1340	Tired Flounders	126
1341	Hasty Exit	123
1342	Plastic Paradise	126

SQL Indexes

- When primary key is declared, DBMS automatically creates unique index
- The **CREATE INDEX** command can be used to create indexes on the basis of any selected attribute
- **UNIQUE** qualifier prevents a value that has been used before
 - Composite indexes prevent data duplication
- To delete an index use the **DROP INDEX** command

SQL Indexes - Examples

Syntax:

```
CREATE [UNIQUE] INDEX indexname  
    ON tablename (col1 [, col2]);
```

Examples:

```
CREATE UNIQUE INDEX P_CODEX  
    ON PRODUCT (P_CODE);  -- Creates index on column P_CODE  
  
CREATE INDEX PROD_PRICEX  
    ON PROD (P_PRICE DESC);  -- Creates index in desc. order  
  
DROP INDEX PROD_PRICEX;  -- Deletes index PROD_PRICEX
```

Additional Data Definition Commands

- **ALTER TABLE** command: To make changes in the table structure
- Keywords used with the command
 - ADD - Adds a column
 - MODIFY - Changes column characteristics
 - DROP - Deletes a column
- Also used to:
 - Add table constraints
 - Remove table constraints

Changing a Column's Data Type and Data Characteristics

- ALTER used to change data type and characteristics
 - Some RDBMSs do not permit changes to data types unless column is empty
 - Changes in characteristics are permitted if they do not alter the existing data type
- Syntax:
 - Data Type: ALTER TABLE *tablename* MODIFY (*columnname*(*datatype*));
 - Data Characteristic: ALTER TABLE *tablename* MODIFY (*columnname*(*characteristic*));

Adding and Dropping Columns

- Adding a column
 - Use ALTER and ADD
 - Do not include the NOT NULL clause for new column
- Dropping a column
 - Use ALTER and DROP
 - Some RDBMSs impose restrictions on the deletion of an attribute

ALTER TABLE – Examples

```
ALTER TABLE PRODUCT
```

```
ADD (P_SALECODE CHAR(1));
```

```
ALTER TABLE PRODUCT
```

```
MODIFY P_SALECODE CHAR(2);
```

```
ALTER TABLE PRODUCT
```

```
DROP COLUMN P_SALECODE;
```

Deleting a Table from the Database

- **DROP TABLE:** Deletes table from database
 - Syntax - DROP TABLE *tablename*;
 - Can drop a table only if it is not the one side of any relationship
 - RDBMS generates a foreign key integrity violation error message if you try to drop a referenced table