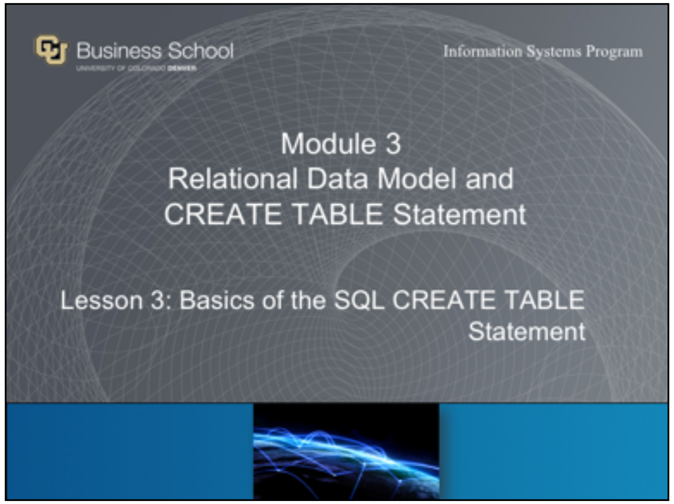


Business School
UNIVERSITY OF COLORADO DENVER

Information Systems Program

Module 3
Relational Data Model and
CREATE TABLE Statement

Lesson 3: Basics of the SQL CREATE TABLE Statement

Welcome to Lesson 3 of Module 3 on the relational data model and the CREATE TABLE statement

- Careful study of the relational data model
- This lesson covers the basic syntax of the relational data model.

Opening question:

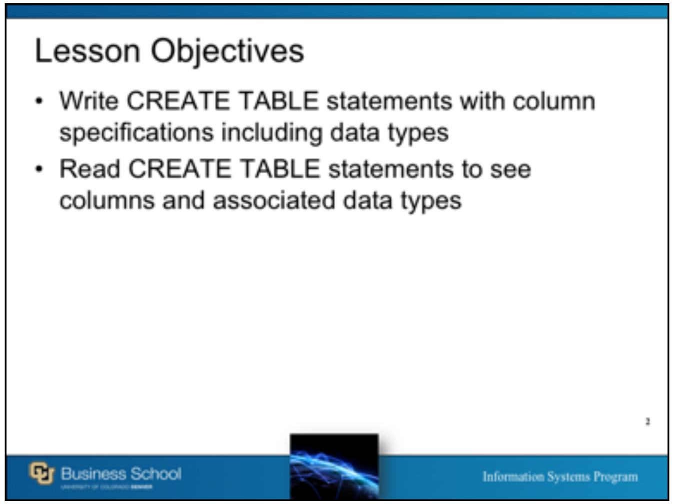
- Why have DBMS vendors created development environment interfaces instead of the CREATE TABLE statement?

Relational databases are the dominant commercial standard

- Simplicity and familiarity with table manipulation
- Strong mathematical framework
- Lots of research and development

Goals of Lesson 3

- Write CREATE TABLE statements with column specifications including data types.
- Read CREATE TABLE statements to see the columns and data types



Lesson Objectives

- Write CREATE TABLE statements with column specifications including data types
- Read CREATE TABLE statements to see columns and associated data types

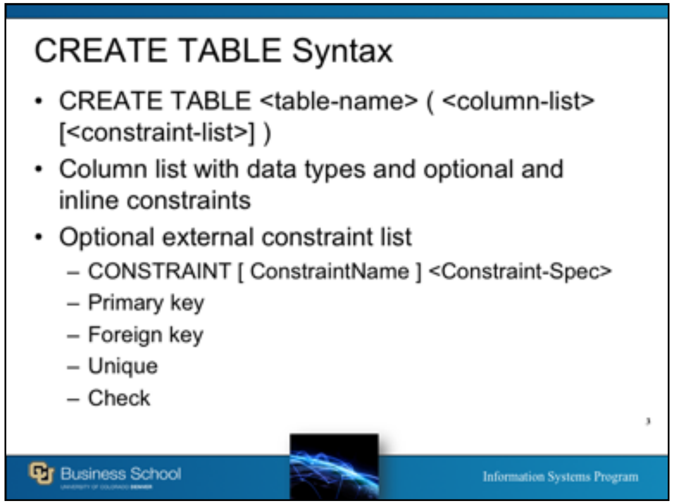
2

Business School

Information Systems Program

Reading and writing simple CREATE TABLE statements

Gaining confidence with syntax requirements



CREATE TABLE Syntax

- CREATE TABLE <table-name> (<column-list> [<constraint-list>])
- Column list with data types and optional and inline constraints
- Optional external constraint list
  - CONSTRAINT [ConstraintName] <Constraint-Spec>
  - Primary key
  - Foreign key
  - Unique
  - Check

3

Business School

Information Systems Program

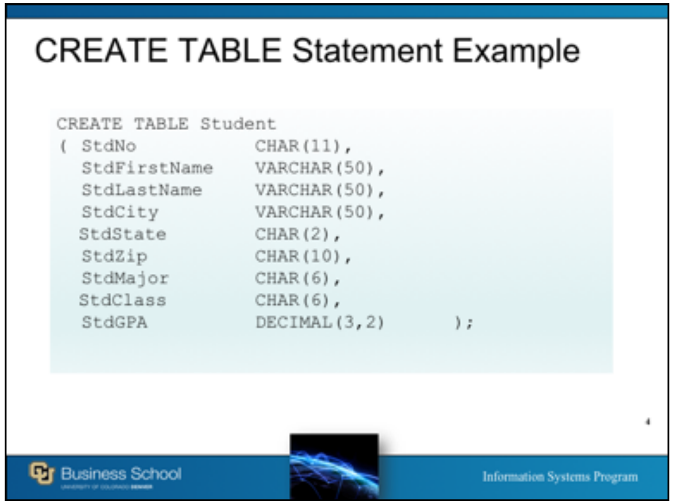
CREATE TABLE keywords followed by parentheses and column list with an optional constraint list.

Column list: list of column definitions separated by commas

- Data type
- Optional default value (DEFAULT keyword and value)
- Optional inline constraint typically NOT NULL

Constraint list:

- CONSTRAINT keyword followed by optional constraint name
- Constraint specification: keyword, column name or condition
- Constraints covered in next lecture



CREATE TABLE Statement Example

```
CREATE TABLE Student
( StdNo           CHAR(11),
  StdFirstName    VARCHAR(50),
  StdLastName     VARCHAR(50),
  StdCity         VARCHAR(50),
  StdState        CHAR(2),
  StdZip          CHAR(10),
  StdMajor        CHAR(6),
  StdClass        CHAR(6),
  StdGPA          DECIMAL(3,2)      );
```

4

Business School

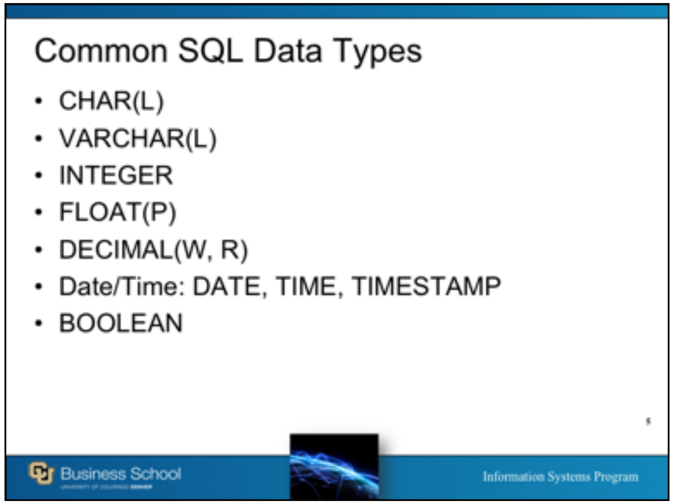
Information Systems Program

Define table name, column names, and column data types

Other clauses added later in the lecture

Data type:

- Set of values
- Permissible values
- Vary by DBMS
- CHAR: fixed length character strings
- VARCHAR: variable length character strings
- DECIMAL: fixed precision numbers
- Table 2-2 lists common data types



Common SQL Data Types

- CHAR(L)
- VARCHAR(L)
- INTEGER
- FLOAT(P)
- DECIMAL(W, R)
- Date/Time: DATE, TIME, TIMESTAMP
- BOOLEAN

5

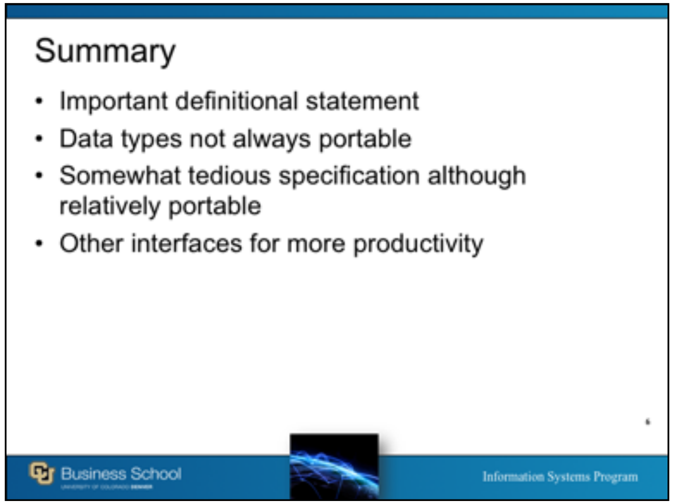
Business School

Information Systems Program

CHAR: fixed length character strings

VARCHAR: variable length character strings

Date/Time: SQL standard provides 3 data types; most DBMSs only support one data type; data type name is not standard across DBMSs



Summary

- Important definitional statement
- Data types not always portable
- Somewhat tedious specification although relatively portable
- Other interfaces for more productivity

6

Business School

UNIVERSITY OF COLORADO DENVER

Information Systems Program

CREATE TABLE statement important because of relative portability

Other interfaces make DBAs more productive