Creating Tables, Defining Constraints

Rose-Hulman Institute of Technology

Curt Clifton

Outline

- Data Types
- Creating and Altering Tables
- Constraints
  - Primary and Foreign Key Constraints
  - Row and Tuple Checks
- Generating Column Values
- Generating Scripts

Data Types

System-supplied Data Types

- Numeric
  - Integer
  - Exact numeric
  - Approximate numeric
  - Monetary
- Date and Time
- Character and Unicode Character
- Binary
- Other

Slide based on MS-CreatingTables.ppt

User-defined Data Types

- Simple, self-documenting short-hand
- Creating:
  - CREATE TYPE ssn
    FROM varchar(11) NOT NULL
- Dropping:
  - DROP TYPE ssn
- Advanced use: C# objects

Guidelines for Data Types

- If Column Length Varies, Use a Variable Data Type
- Use tinyint Appropriately
- For Numeric Data Types, Commonly Use decimal
- Use money for Currency
- Do Not Use float or real as Primary Keys

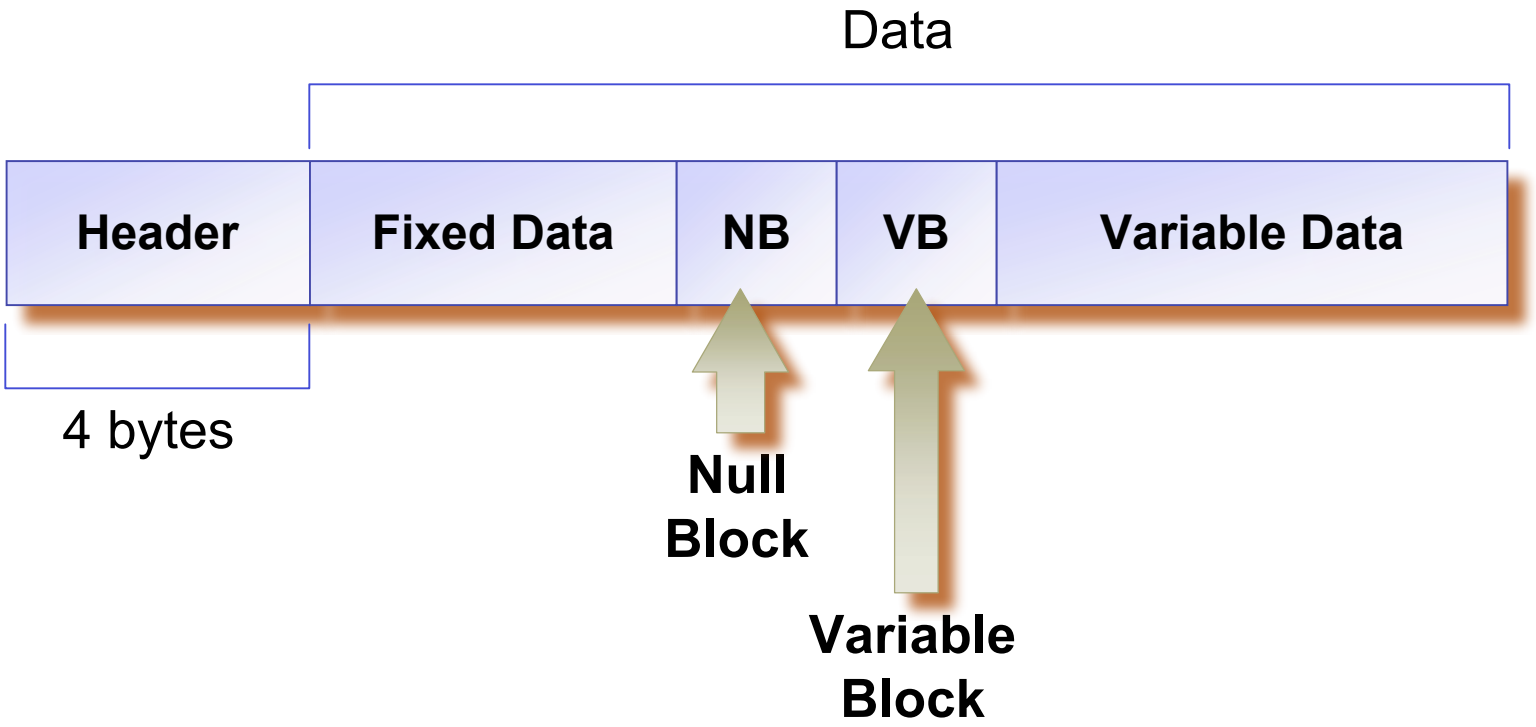
Slide based on MS-CreatingTables.ppt

Creating and Altering Tables

Creating Tables

- Need:
  - Table name
  - Column names and types
- Basic Example:
  - CREATE TABLE Soda(
    name CHAR(20),
    manf CHAR(20)
    );

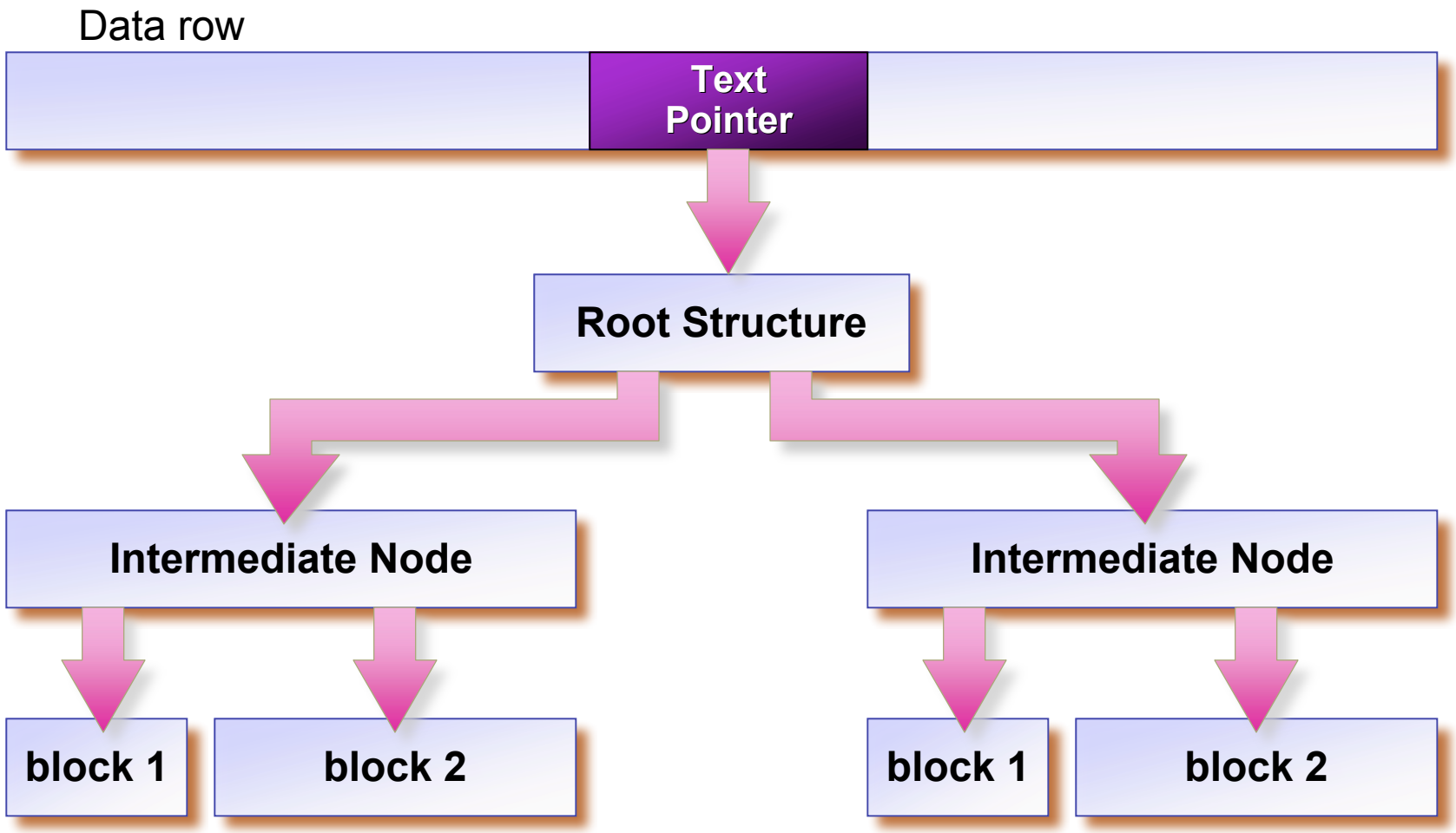
How SQL Server Organizes Data



A Single Data Row

Slide based on MS-CreatingTables.ppt

Big @$$ Data



Slide based on MS-CreatingTables.ppt

Altering Tables

- Adding columns:
  - ALTER TABLE Soda
    ADD msrp float;
- Changing columns:
  - ALTER TABLE Soda
    ALTER COLUMN msrp money;
- Dropping columns:
  - ALTER TABLE Soda
    DROP COLUMN manf;

Dropping Tables

- DROP TABLE Soda;

Constraints

- A requirement on data elements or the relationship between data elements that the DBMS is required to enforce

Kinds of Constraints

- Primary keys (entity integrity)
- Foreign keys (referential integrity)
- Attribute-based
  - Restrictions on the value of a single attribute
- Row-based
  - Restrictions on the value of one attribute in row based on value of other attributes
- Assertions
  - Later…

Specifying Primary Key Constraint

- Examples:
  - CREATE TABLE Soda (
    name CHAR(20) PRIMARY KEY,
    manf CHAR(20)
    );
  - CREATE TABLE Likes(
    customer CHAR(30),
    soda CHAR(20),
    PRIMARY KEY(customer, soda)
    );

Foreign Key Constraints

- Consider foreign keys in Sells relation…

Specifying Foreign Key Constraints

- CREATE TABLE Sells(
  rest CHAR(20) REFERENCES Rest(name),
  soda CHAR(20) REFERENCES Soda(name),
  price money); *or*
- CREATE TABLE Sells(
  rest CHAR(20),
  soda CHAR(20),
  price money,
  FOREIGN KEY(rest) REFERENCES Rest(name),
  FOREIGN KEY(soda) REFERENCES Soda(name));

Foreign Key Restriction

- Referenced attributes must be either:
  - PRIMARY KEY or else
  - UNIQUE (another element constraint)

Enforcing Foreign-Key Constraints

- What changes to the SodaBase data might break referential integrity?

Change to Table with Foreign Key

- How should we handle an insert or update to the table with the foreign key that would break referential integrity?

Change to Table with Primary Key

- How should we handle an update or delete to the table with the primary key that would break referential integrity?

3 Solutions to Primary Key Change

- Reject!
  - This is the default
- Cascade
  - Make same change to foreign key
- Set null
  - Set foreign key to null

Example: Default Policy

- Suppose ‘Coke’ is referenced by Sells…
  - We attempt to delete ‘Coke’ from Soda table
    - Rejected!
  - We attempt to update ‘Coke’ row, changing ‘Coke’ to ‘Coca-Cola’
    - Rejected!
- Forces Sells table to be changed first

Example: Cascade Policy

- Suppose we delete Coke row from Soda
  - Then automatically delete all rows for Coke from Sells
- Suppose we update the Coke row, changing ‘Coke’ to ‘Coca-Cola’
  - Then automatically change all rows in Sells referencing Coke to reference Coca-Cola instead

Example: “Set Null” Policy

- Suppose we delete Coke row from Soda
  - Then automatically change all rows referencing Coke in Sells to have nulls
- Suppose we update the Coke row, changing ‘Coke’ to ‘Coca-Cola’
  - Then automatically change all rows in Sells referencing Coke to have nulls

Choosing a Policy

- Can independently choose policy…
  - For update
  - For delete
- What policy should we use for…
  - Deleting soda? Why?
  - Updating soda name? Why?

Specifying a Policy

- Follow foreign-key declaration with:
  - [ON UPDATE {SET NULL | CASCADE}]
    [ON DELETE {SET NULL | CASCADE}]
- Omitted clause means default policy

Example

- CREATE TABLE Sells(
  rest CHAR(20) REFERENCES Rest(name)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
  soda CHAR(20) REFERENCES Soda(name)
  ON DELETE SET NULL
  ON UPDATE CASCADE,
  price money
  );

Attribute-based Checks

- Can constrain single attribute values
- Syntax:
  - CHECK(*condition*)
- Condition can use:
  - Name of checked attribute
  - Subqueries
- Checked only upon insertion, update

Example

```
CREATE TABLE Customer(
    name        CHAR(20) PRIMARY KEY,
    addr        CHAR(50),
    phone       CHAR(8)
      CHECK (phone LIKE
            '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]')
);
```

Same or Different?

- CREATE TABLE Sells (
  rest CHAR(20),
  soda CHAR(20)
  REFERENCES
  Soda(name),
  price money
  );

- CREATE TABLE Sells (
  rest CHAR(20),
  soda CHAR(20)
  CHECK (
  soda IS NULL
  OR soda IN
  (SELECT name
  FROM Soda)),
  price money
  );

Row-Based Checks

- Can also put CHECK at end of table declaration
- Can reference any attribute in table
- CHECK for each tuple…
  - Inserted or
  - Updated

Example

- Only Joe's can sell Coke for more than $2
- CREATE TABLE Sells (

  rest CHAR(20),

  soda CHAR(20),

  price money,

  CHECK(*condition*)

  );
- What should *condition* be?

Generating Column Values

- Table identity columns
- Globally unique identifiers

Table Identity Column

- Constraint on single column of table
- Column must be integer or decimal data type
- Syntax:
  - IDENTITY [(*seed*, *increment*)]
- Example:
  - CREATE TABLE Users(
    name CHAR(20),
    id int IDENTITY (0, 5));

Getting Last Identity Value

- Use @@identity in scripts
- INSERT INTO Users(name)
  VALUE ('Molly');

```
SELECT 'Last identity used: ' +
            CONVERT(char, @@identity)
    AS Answer;
```

GUIDs

- Globally unique identifiers
- Generated with newid() function
- Used with DEFAULT constraint

Example

- CREATE TABLE Household(
  HouseholdID uniqueidentifier
  NOT NULL DEFAULT newid(),
  …
  );

Generating Scripts

- Can generate scripts from objects
  - Right click database
  - Tasks → Generate Scripts…
- Useful for:
  - Storing schemas in version control system
  - Creating test environment
  - Training

Recommended Practices

- Specify Appropriate Data Types and Data Type Sizes (duh!)
- Always Specify Column Characteristics in CREATE TABLE
- Generate Scripts to Recreate Database Objects

Slide based on MS-CreatingTables.ppt