Relational Databases: Database Creation using SQL

ACS 575: Database Systems

Instructor: Dr. Jin Soung Yoo, Professor
Department of Computer Science
Purdue University Fort Wayne

References

- R. Elmasri et al., Fundamentals of Database Systems, 7th ed. Ch 6.
- W. Lemanhieu, et al., Principles of Database Management, Ch 7.

Outline

- Table Creation
  - Attribute Types
  - Data Definition Language (DDL) in SQL
  - Create Table
  - Drop/Alter Table

Designing Fields

- Field: smallest unit of application data recognized by system software
- Field design
  - Choosing data type
  - Controlling data integrity
  - Coding, compression, encryption

Data Types Commonly Used in Oracle

TABLE 5-1 Commonly Used Data Types in Oracle 11g

Data Type	Description
VARCHAR2	Variable-length character data with a maximum length of 4,000 characters; you must enter a maximum field length [e.g., VARCHAR2(30) specifies a field with a maximum length of 30 characters]. A string that is shorter than the maximum will consume only the required space.
CHAR	Fixed-length character data with a maximum length of 2,000 characters; default length is 1 character (e.g., CHAR(5) specifies a field with a fixed length of 5 characters, capable of holding a value from 0 to 5 characters long).
CLOB	Character large object, capable of storing up to (4 gigabytes – 1) * (database block size) of one variable-length character data field (e.g., to hold a medical instruction or a customer comment).
NUMBER	Positive or negative number in the range 10^{-130} to 10^{126}; can specify the precision (total number of digits to the left and right of the decimal point) and the scale (the number of digits to the right of the decimal point). For example, NUMBER(5) specifies an integer field with a maximum of 5 digits, and NUMBER(5,2) specifies a field with no more than 5 digits and exactly 2 digits to the right of the decimal point.
DATE	Any date from January 1, 4712 B.C., to December 31, 9999 A.D.; DATE stores the century, year, month, day, hour, minute, and second.
BLOB	Binary large object, capable of storing up to (4 gigabytes – 1) * (database block size) of binary data (e.g., a photograph or sound clip).

SQL Data Types

ANSI SQL data type	Oracle data type	MS SQL Server data type	IBM DB2 data type	MySQL Data type
CHARACTER(n) CHAR(n)	CHAR(n)	CHARACTER(n) CHAR(n)	CHARACTER(n)	CHAR(n)
CHARACTER VARYING(n) CHAR VARYING(n)	VARCHAR2(n), VARCHAR(n)	VARCHAR(n)	VARCHAR(n)	VARCHAR(n)
NATIONAL CHARACTER(n) NATIONAL CHAR(n) NCHAR(n)	CHAR(n*2) NCHAR(n)	NCHAR(n)		
NATIONAL CHARACTER VARYING(n) NATIONAL CHAR VARYING(n) NCHAR VARYING(n)	NVARCHAR2(n) VARCHAR2(n*2)	NVARCHAR(n)		
NUMERIC[(p,s)] DECIMAL[(p,s)]	NUMBER(p,s)		DECIMAL(p,s)	NUMERIC
INTEGER INT SMALLINT	NUMBER NUMBER NUMBER(6)	INTEGER SMALLINT	INTEGER SMALLINT	INTEGER INT SMALLINST
FLOAT DOUBLE PRECISION REAL	NUMBER NUMBER REAL - FLOAT(63)	FLOAT	FLOAT	DECIMAL, FLOAT DOUBLE REAL

More SQL Data Types …

ANSI SQL data type	Oracle data type	MS SQL Server data type	IBM DB2 data type	MySQL Data type
	LONG		LONG VARCHAR(n)	
	CLOB	TEXT		LONGTEXT MIDIUMTEXT
	BLOB	IMAGE		LOGBLOB MEDIUMBLOB
	DATE	DATETIME		DATE, DATETIME, TIME, TIMESTAMP
	NUMBER	TIMESTMP		
	NUMBER			YEAR
	RAW			BIT
	LONG RAW			
	ROWID			
	NUMBER(19,4)	MONEY		
	NUMBER (1)	BIT		

Data Types of DBMS

- Oracle data type
  - http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/sql_elements001.htm#i45441 (Table 2-1 Oracle Built-in DataTypes)
  - https://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm
- MS SQL Server data type

  http://msdn.microsoft.com/en-us/library/ms187752.aspx

  http://docs.oracle.com/cd/E10405_01/doc/appdev.120/e10379/ss_oracle_compared.htm#i1026427
- MySQL data type

  http://dev.mysql.com/doc/refman/5.6/en/data-types.html

  http://docs.oracle.com/cd/E10405_01/doc/appdev.120/e10380/oracle_mysql_compared.htm#BABDFGBD

SQL (Structured Query Language)

- Structured Query Language
- Developed by IBM (system R) in the 1970s
- The standard for relational database management systems (RDBMS)
- History of SQL
  - 1970–E. Codd develops relational database concept
  - 1974-1979–System R with Sequel (later SQL) created at IBM Research Lab
  - 1979–Oracle markets first relational DB with SQL
  - 1986–ANSI SQL standard released
  - 1989, 1992, 1999, 2003–Major ANSI standard updates
  - Current–SQL is supported by most major database vendors

SQL (Conti)

- SQL can divided to categories.
  - DML (Data Manipulation Language)
    - Commands that maintain and query a database
      - Query (SELECT)
      - Data managements (INSERT, DELETE, UPDATE)
  - DDL (Data Definition Language)
    - Commands that define a database, including CREATE, ALTER, and DROP tables or other objects (e.g., view, user, synonym, sequence, role, …)
  - DCL (Data Control Language)
    - Commands that control a database, including administering privileges and committing data (e.g., , GRANT, REVOKE, COMMIT, ROLLBACK….)

DDL- CREATE TABLE

- Create a new table in the database. The table name must not already exist. CREATE TABLE has the following

```
CREATE TABLE <table_name>
( <column1_name> <data type>  [<[not] null>|default|CHECK],
  <colum1n_name> <data type> [<[not] null>|default|CHECK],
    …,
    [constraint <name>] primary key | foreign key | unique
);
```

- An alternate syntax can be used to create a table with a subset of rows or columns from an existing table.

```
CREATE TABLE  <table_name>
 AS
<sql select statement>;
```

Table Creation with Query Result

- Syntax

  CREATE TABLE <table_name>

  AS

  <sql select statement>;

- Example

  CREATE TABLE EmployeeOfDept10

  AS

  SELECT *

  FROM Employee

  Where Dno=10;

CREATE TABLE statement

```
CREATE TABLE tablename
( {column definition   [table constraint] } . , . .
[ON COMMIT {DELETE | PRESERVE} ROWS] );

where column definition ::=
column_name
        {domain name | datatype [(size)] }
        [column_constraint_clause . . .]
        [default value]
        [collate clause]

and table constraint ::=
        [CONSTRAINT constraint_name]
        Constraint_type [constraint_attributes]
```

1. Identify data types for attributes
2. Identify columns that can and cannot be null
3. Identify columns that must be unique (candidate keys)
4. Identify primary key–foreign key mates
5. Determine default values
6. Identify other constraints on columns (e.g., check constraint for domain specifications)

EXAMPLE
(Oracle syntax)

```
CREATE TABLE Customer_T
                    (CustomerID                    NUMBER(11,0)        NOT NULL,
                     CustomerName                  VARCHAR2(25)        NOT NULL,
                     CustomerAddress               VARCHAR2(30),
                     CustomerCity                  VARCHAR2(20),
                     CustomerState                 CHAR(2),
                     CustomerPostalCode            VARCHAR2(9),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID));

CREATE TABLE Order_T
                   (OrderID                        NUMBER(11,0)        NOT NULL,
                    OrderDate                      DATE DEFAULT SYSDATE,
                    CustomerID                     NUMBER(11,0),
CONSTRAINT Order_PK PRIMARY KEY (OrderID),
CONSTRAINT Order_FK FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID));

CREATE TABLE Product_T
                   (ProductID                      NUMBER(11,0)        NOT NULL,
                    ProductDescription             VARCHAR2(50),
                    ProductFinish                  VARCHAR2(20)
                                 CHECK (ProductFinish IN ('Cherry', 'Natural Ash', 'White Ash',
                                                          'Red Oak', 'Natural Oak', 'Walnut')),
                    ProductStandardPrice           DECIMAL(6,2),
                    ProductLineID                  INTEGER,
CONSTRAINT Product_PK PRIMARY KEY (ProductID));

CREATE TABLE OrderLine_T
                   (OrderID                        NUMBER(11,0)        NOT NULL,
                    ProductID                      INTEGER             NOT NULL,
                    OrderedQuantity                NUMBER(11,0),
CONSTRAINT OrderLine_PK PRIMARY KEY (OrderID, ProductID),
CONSTRAINT OrderLine_FK1 FOREIGN KEY (OrderID) REFERENCES Order_T(OrderID),
CONSTRAINT OrderLine_FK2 FOREIGN KEY (ProductID) REFERENCES Product_T(ProductID));
```

Defining Attributes and their Data Types

```
CREATE TABLE Product_T
                    (ProductID                  NUMBER(11,0),
                     ProductDescription         VARCHAR2(50),
                     ProductFinish              VARCHAR2(20),
                     ProductStandardPrice       DECIMAL(6,2),
                     ProductLineID              INTEGER,
CONSTRAINT Product_PK PRIMARY KEY (ProductID));
```

Controlling the Values in Attributes

```
CREATE TABLE Order_T
                (OrderID                    NUMBER(11,0)        NOT NULL,
                 OrderDate                  DATE DEFAULT SYSDATE,
                 CustomerID                 NUMBER(11,0),
                  . . .
                                                                );
CREATE TABLE Product_T
                (ProductID                  NUMBER(11,0)        NOT NULL,
                 ProductDescription         VARCHAR2(50),
                 ProductFinish              VARCHAR2(20)
                 CHECK (ProductFinish IN ('Cherry', 'Natural Ash', 'White Ash',
                                          'Red Oak', 'Natural Oak', 'Walnut')),
                 ProductStandardPrice       DECIMAL(6,2),
                 ProductLineID              INTEGER,
                  . . .
                                                                );
```

default value

Domain CHECK constraint

Simple Primary Key Specification

```
CREATE TABLE Product_T
                    (ProductID                  NUMBER(11,0),
                     ProductDescription         VARCHAR2(50),
                     ProductFinish              VARCHAR2(20),
                     ProductStandardPrice       DECIMAL(6,2),
                     ProductLineID              INTEGER,
CONSTRAINT Product_PK PRIMARY KEY (ProductID));
```

- A DBMS implicitly specifies that primary key attribute has "not null" constraints even if you does not specify NOT NULL in the attribute.
- Constraint names are recommended!! Otherwise, DBMS gives an arbitrary name.

Composite Primary Key Specification

```
CREATE TABLE OrderLine_T
                (OrderID                    NUMBER(11,0)      NOT NULL,
                 ProductID                  INTEGER           NOT NULL,
                 OrderedQuantity            NUMBER(11,0),
CONSTRAINT OrderLine_PK PRIMARY KEY (OrderID, ProductID),
                . . .
                                                              );
```

- Some primary keys are composite – composed of multiple attributes.

Candidate Key Specification

- Possibly many *candidate keys* (specified using UNIQUE), one of which is chosen as the *primary key*.

```
CREATE TABLE Students
        (sid CHAR(20),
         name CHAR(20),
         login CHAR(10),
         age INTEGER,
         gpa REAL,
         CONSTRAINT PK_students PRIMARY KEY (sid),
         CONSTRAINT unique1 UNIQUE(login)
);
```

Foreign Key Specification

```
CREATE TABLE Customer_T
                    (CustomerID                 NUMBER(11,0)        NOT NULL,
                     CustomerName               VARCHAR2(25)        NOT NULL,
                     CustomerAddress            VARCHAR2(30),
                     CustomerCity               VARCHAR2(20),
                     CustomerState              CHAR(2),
                     CustomerPostalCode         VARCHAR2(9),
CONSTRAINT Customer_PK PRIMARY KEY (CustomerID));

CREATE TABLE Order_T
                   (OrderID                     NUMBER(11,0)        NOT NULL,
                    OrderDate                   DATE DEFAULT SYSDATE,
                    CustomerID                  NUMBER(11,0),
CONSTRAINT Order_PK PRIMARY KEY (OrderID),
CONSTRAINT Order_FK FOREIGN KEY (CustomerID) REFERENCES Customer_T(CustomerID));
```

Referential Integrity in SQL

- SQL/92 and SQL:1999 support all 4 options on deletes and updates
  - Default is NO ACTION (*delete/update is rejected*)
  - CASCADE
  - SET NULL / SET DEFAULT
  - Example

```
CREATE TABLE Students
( sid CHAR(20),
  student_name CHAR(20),
  CONSTRAINT pk_student
                PRIMARY KEY (sid));
```

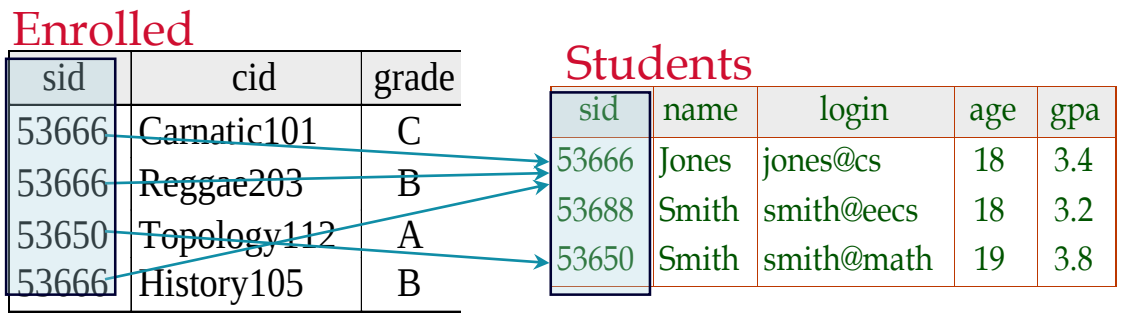
```
CREATE TABLE Enrolled
(    sid CHAR(20) DEFAULT '999',
     cid CHAR(20),
     grade CHAR(2),
    CONSTRAINT pk_enrolled PRIMARY KEY (sid,cid),
    CONSTRAINT pk1_enrolled
        FOREIGN KEY (sid)  REFERENCES Students (Sid)
             ON DELETE CASCADE
             ON UPDATE SET DEFAULT
 );
```

Referential Integrity Violations By DELETE operation

- If the primary key value of the tuple being deleted is referenced from other tuples in the database
- The violation can be remedied by several actions.
  - RESTRICT option: reject the deletion
  - CASCADE option: also delete all tuples refer to the delete tuple.
  - SET NULL / SET DEFULT option: set the foreign keys of the referencing tuples to NULL/default

Example

- What should be done in Enrolled tuples if a Students tuple is deleted?
  - Also delete all Enrolled tuples that refer to it. (CASCADE)
  - Disallow deletion of a Students tuple that is referred to. (RESTRICT)
  - Set sid in Enrolled tuples that refer to it to a *default sid*. (SET DEFAULT)
  - Set sid in Enrolled tuples that refer to it to a special value *null*, denoting *`unknown'* or *`inapplicable'*. (SET NULL)



Referential Integrity Violations By UPDATE operation

- If the primary key value of the tuple being updated is referenced from other tuples in the database
- The violation can be remedied by several actions.
  - RESTRICT option: reject the deletion
  - CASCADE option: also update all foreign key values refer to the updated field.
  - SET NULL / SET DEFULT option: set the foreign keys of the referencing tuples to NULL/default

Outline

- Table Creation in Database Management Systems (A part of Ch.4)
  - Attribute Types
  - Data Definition Language (DDL) in SQL
  - Create Table
  - ☞ Drop/Alter Table

Destroying Relations

- Destroys the relation Students. The schema information *and* the tuples are deleted.

  DROP TABLE Students

- If there are referential integrity constraints among tables, the order of drop is important.
  - E.g., DROP TABLE enrolled;

    then, DROP TABLE students;

Changing Relations

- ALTER TABLE statement allows you to change column specifications:

  ALTER TABLE table_name alter_table_actions

- Some table Actions:

  ADD [COLUMN] column_definition

  ALTER [COLUMN] column_name SET DEFAULT default-value

  ALTER [COLUMN] column_name DROP DEFAULT

  DROP [COLUMN] column_name [RESTRICT][CASCADE]

  ADD table_constraint

Examples

- Adding a new column

  ALTER TALBE Students
  ADD (FirstYear number(4))

  - The schema of Students is altered by adding a new field; every tuple in the current instance is extended with a *null* value in the new field

- Modifying a column

  ALTER TALBE Students
  MODIFY (FirstYear number(2));

  - If the column has no value, the column type can be changed and the column length can be decreased

Examples

- Renaming a column

```
ALTER TALBE Students
RENAME COLUMN FirstYear TO FYear ;
```

- Dropping a column

```
ALTER TALBE Students
DROP (FYear) ;
```

- Enable/Disable a constraint

```
ALTER TALBE Students
DISABLE PRIMARY KEY;
```

```
ALTER TALBE Students
DISABLE a constraint_name;
```

Examples

- Drop a constraint

  ALTER TALBE Students
  DROP a constraint_name;

- Add a constraint

  ALTER TALBE Students
  ADD (CONSTRAINT FK2_Dept FOREIGN KEY (dept) REFERENCES DEPARTMENT)

- Many others…
  - For Oracle cases, see http://download.oracle.com/docs/cd/B28359_01/server.111/b28286/statements_3001.htm

Adding, Deleting and Updating Tuples

- Can insert a single tuple using:

```
INSERT INTO Students (sid, name, login, age, gpa)
VALUES (53688, 'Smith', 'smith@ee', 18, 3.2);
```

- Can delete all tuples satisfying some condition (e.g., name = Smith):

```
DELETE  FROM Students;

DELETE  FROM Students
WHERE name = 'Smith';
```

- Can modify columns in a tuple using:

```
UPDATE Students
SET   age = age + 1;

UPDATE Students
SET   age = age + 1
WHERE S.sid = 53688;
```

Powerful variants of these commands are available; more later!