

SQL: DDL

John Ortiz

Cs.utsa.edu

SQL Data Definition Language

- ◆ Used by DBA or Designer to specify schema
- ◆ A set of statements used to define and to change the definition of tables, columns, data types, constraints, views, indexes, ...
- ◆ SQL DDL & DML are integrated.
 - ▲ A DDL statement often needs to contain some DML statements.

A Sample University Schema

- ◆ Students(SID:string, Name:string, Age:integer, Sex:char, Major:string, GPA:real)
- ◆ Courses(Cno:string, Title:string, Hours:integer, Dept:string)
- ◆ Enrollment(SID:string, Cno:string, Year:string, Grade:string)
- ◆ Offers(Cno:string, Year:integer, FID:string)
- ◆ Faculty(FID:string, Name:string, Rank:string, Dept:string, Salary:real)
- ◆ Departments(Name:string, Location:string, ChairID:string)

Create Students Table

- ◆ In SQL*Plus:

```
SQL> create table Students
```

```
2 (SID char(9) not null,
```

```
3 Name varchar2(25),
```

```
4 Age integer,
```

```
5 Sex char(1),
```

```
6 Major char(4),
```

```
7 GPA number(3,2),
```

```
8 primary key (SID));
```

Create Tables Syntax

```
create table Table-Name (  
  Col-Name Type Deft-Val Col-Constraint,  
  ...  
  Col-Name Type Deft-Val Col-Constraint,  
  Table-Constraint,  
  ...  
  Table-Constraint);
```

Oracle SQL Built-in Data Types

- ◆ char(n). String of $n < 2000$ char
 - ◆ varchar2(n). String up to $n \leq 4000$ char
 - ◆ long. Char string of length up to 2GB
 - ◆ number(n,m). n digits, m after decimal point.
 - ◆ number. Integer or real up to 40 digits
 - ◆ integer. Integer up to 40 digits
 - ◆ blob. Binary data up to 4 GB
 - ◆ date. DD-MMM-YY
 - ◆ time. HH:MM:SS
- ◆ These may differ from SQL2 & SQL-1999.

SQL Integrity Constraints

- ◆ Rules or regulations imposed to ensure data integrity.
 - ▲ Column Constraints.
 - ▲ Table Constraints.
 - ▲ Assertions (Multiple-table Constraints).
 - ▲ Triggers.
 - ▲ Primary Key, Foreign Key, Check, Not Null, Unique, ...

Column Definition

- ◆ Syntax for column definition:

col_name data_type [default value] [column constraints]

- ◆ Syntax for column constraints:

[constraint constraint_name]

[not] null | check condition |

unique | primary key |

references table_name [(column)]

[on delete cascade]

Column Constraints

- ◆ not null. Can not take null value.
 - ◆ unique. Can not have identical non-null values
 - ◆ primary key. Both not null and unique
 - ◆ references T(A). All non-null values must be currently in T.A.
 - ◆ check (condition). Values must satisfy the check condition.
- Can be expressed as table constraints, too.

Column Constraints Example

```
SQL> create table Courses  
  (CNo char(6) primary key,  
   Title varchar2(50) not null,  
   Hours integer default 3  
   check (Hours > 0 and hours < 6),  
   Dept varchar2(20)  
   references Departments(Name));
```

Table Constraints

- ◆ Syntax for table constraints:

[constraint constraint_name]

check condition |

unique (column {, column}) |

primary key (column {, column}) |

foreign key (column {, column})

references table_name[(column {, column})]

[on delete cascade]

Table Constraints Example

```
SQL> create table Enrollment  
(SID char(9) not null references Students,  
CNo varchar2(7) not null,  
Year number(2) not null,  
Grade char(2),  
primary key (SID, CNo, Year),  
foreign key (CNo) references Courses);
```

Table Constraints Example (cont.)

```
SQL> create table Students  
(SID char(9) primary key,  
Name varchar2(25),  
Age integer check(Age > 18 and Age < 100),  
Sex char check(Sex in {'F', 'M'}),  
Major varchar2(4)  
GPA number (3,2) not null,  
constraint ic12 check (GPA >= 2.0 and  
                  (Major = 'IS' or GPA >= 3.0)));
```

Referential Integrity & Data Update

- ◆ Assume that `Courses.Dept` references `Departments.Name`. What should the system do to students if we change a department's name or delete a department?
- ◆ SQL provides four options:
 - ▲ No action. Disallow such an update.
 - ▲ Cascade. Accept update and update all affected foreign key values.
 - ▲ Set default. Accept update & set default FK.
 - ▲ Set null. Accept update & set FK to null.

Referential Integrity Example

```
SQL>create table Courses (  
  CNo char(6) not null primary key,  
  Title varchar(35) not null,  
  Hours int check (Hours between 1 and 5),  
  Dept varchar(20),  
  foreign key (Dept) references  
    Departments(Name)  
  on delete no action on update cascade);
```

Drop Table

- ◆ Delete schema definition of a table.

`drop table Table-Name;`

- ◆ Problem:

`drop table Departments`

will fail if it is referenced by foreign keys.

- ◆ Solution:

`drop table Departments cascade constraints;`

All referential constraints will be dropped before the table is dropped.

Alter Table

- ◆ Change table schema (even after entering data)

- ◆ Add a new column.

```
alter table Students  
add (Address varchar2(40));
```

- ◆ Add a new constraint.

```
alter table Students add (unique(Address));
```

- ◆ Modify a column definition.

```
alter table Students  
modify (Name varchar2(30));
```

Alter Table (cont.)

- ◆ Remove a column.

alter table Students drop (Address);

- ◆ Enable and disable a constraint on a table

alter table Students enable constraint ic12;

alter table Students disable constraint ic12;

- ☛ Newly added column can not be specified as not null.
- ☛ Can not modify a column to a type of a smaller size.

Simple Update Statements

- ◆ Insert Statement:

```
insert into table_name [(column {, column})]  
    [values (expression {, expression})]
```

- ◆ Update Statement:

```
update table_name [corr_name]  
    set column = {expression | null}  
    {, column = {expression | null}}  
    [where search_condition]
```

- ◆ Delete Statement:

```
delete from table_name
```

Example of Update

insert into Students

values ('123456789', 'Kathy', 26, 'F', 'CS', null)

or

insert into Students (Name, SID, Age, Major, Sex)

values ('Kathy', '123456789', 26, 'CS', 'F')

Example of Update (cont.)

- ◆ Increase the GPA of the student with SID = 123456789 by 0.5.

```
update Students
```

```
set GPA = GPA + 0.5
```

```
where SID = '123456789'
```

- ◆ Delete all tuples from Students.

```
delete from Students
```

- ☛ The schema of Students remains.

Data Dictionary

- ◆ Data dictionary (system catalog) contains information about all database objects (tables, views, indexes, sequences, etc).
- ◆ Common Oracle Data Dictionary Tables
 - ▲ user_objects(object_name, object_id, object_type, created, last_ddl_time, timestamp, status)
 - ▲ Example database objects include tables, views, sequences, indexes, and packages.

Data Dictionary (cont.)

- ▲ `user_tables(table_name, tablespace_name, num_rows, blocks, empty_blocks, avg_row_len)`
- ▲ `user_tab_columns(name, table_name, column_name, data_type, data_length, nullable, column_id, default_length, data_default, num_distinct, low_value, high_value)`
- ☛ Use `select *` from dictionary to see all system tables and views.
- ☛ Use `describe table-name` to view a schema.

DDL Summary

- ◆ Specify appropriate data type for each column. You may also define your own domains.
- ◆ Specify as many constraints as needed for applications.
- ◆ Specify desirable actions for foreign key constraints.
- ◆ Not all constraints can be specified at the same time. It is necessary to update schemas.
- ◆ Major schema change after data is entered is very costly.