

Database Design and SQL

Chapter 7: Create/Modify Table Structure

CPCM, FSDM

Sagara Samarawickrama

2023S

Chapter Objectives

- Add new rows to the tables with INSERT statement
- Update rows in a table with the UPDATE statement
- Delete rows from a table with the DELETE statement
- Insert rows with DEFAULT values
- Insert rows with NULL Values

Physical Data Modelling

Database Build

once the relations data model is complete, the database is ready to be build or implemented. The steps for database build are identified in following figure

Step	Task
1.	Create DDL statements
2.	Generate database
3.	Implement constraints
4.	Generate test data
5.	Perform database testing

DDL (Data definition Language) commands used to create and modify the structure of database objects

Physical Data Modelling

Command	
CREATE	Create database and objects (table, view, index, and others)
DROP	Delete objects from the database
ALTER	Alter the structure of the database
TRUNCATE	Remove all rows (records) from a table
COMMENT	Add comments to the data dictionary
RENAME	Rename an object existing in the database

DDL (Data definition Language) commands used to create and modify the structure of database objects

Physical Data Modelling

Creating Tables:

A table is a database object that is used to **store data**. Tables are stored in schemas and are uniquely identified by their names. Before data can be entered into table and processed the layout of the table must be described and created. This includes defining the characteristics of **columns**, including **column names**, **column sizes**, the **type of data to be stored in each column**, **valid values the columns can contain**, and so on.

Column description	Data type	Size, digits	SQL name
Employee ID (PK)	INTEGER	4	employee_id
Store ID	INTEGER	4	store_id
First Name	VARCHAR	15	first_name
Middle Initial	VARCHAR	1	m_initial
Last Name	VARCHAR	15	last_name
Hire Date	DATE	10	hire_date
Department ID	INTEGER	4	department_id
Hourly Rate	DECIMAL	5,2	hourly_rate
Hours Worked	DECIMAL	3,1	hours_worked
Sales	DECIMAL	7,0	sales

Physical Data Modelling

Creating Tables:

A table is a database object that is used to **store data**. Tables are stored in schemas and are uniquely identified by their names. Before data can be entered into table and processed the layout of the table must be described and created. This includes defining the characteristics of columns, including column names, column sizes, the type of data to be stored in each column, valid values the columns can contain, and so on.

Physical Data Modelling

Creating Table Command:

The create table command is used to create a table. To create a table, enter the keywords CREATE TABLE followed by the name of the table being created. Following the table name are the column names and their data types. Additional options for the CREATE TABLE command *will be discussed later*

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

```
CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);
```

- The column parameters specify the names of the columns of the table.
- The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

Physical Data Modelling

The rules for naming tables and columns are as follows:

- Names must start with a letter (A–Z).
- After position 1, names can contain letters, numbers, and underscores (_).
- Names cannot exceed 30 characters.
- Names cannot contain spaces.
- Do not use any SQL reserved keywords, such as "select," "create," or "insert," as names for tables or column names.

SQL is a free format language and statement can occupy multiple lines and include blanks between words. **SQL is not case sensitive**

Physical Data Modelling

DROP(DELETE) Table Command:

If it becomes necessary to delete a table , the DROP command is used. To delete a table , the keywords DROP TABLE are entered followed by the name of the table that is to be dropped;

```
DROP TABLE employees;
```

Dropping a table also deletes the data stored in the table.

ALTER TABLE Command:

The ALTER TABLE command alters the definition of a table. The Following are some ways to alters a table definition

ADD NEW Column:



```
ALTER TABLE table_name  
ADD column_name datatype(length);
```

Physical Data Modelling

DROP Column:



```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

Change Column Data Type



Oracle & MySQL

```
ALTER TABLE table_name  
MODIFY column_name datatype(length);
```

TRUNCATE Table Command

The TRUNCATE TABLE command deletes the data inside a table, but not the table itself. In the following example, all data is deleted from the CUSTOMERS table, but the table remains as an empty object

```
TRUNCATE TABLE customers;
```

Physical Data Modelling

DML (Data Manipulation Language):

One group of SQL statements called DML , is used to select and modify data in a database. DML statements are listed below

DML Statement	Description
SELECT	Retrieves data from a database
INSERT	Insert one or more rows into a database table
UPDATE	Updates one or more existing rows in a database table
DELETE	Delete rows from a database table
MERGE	Select rows from one or more tables for update or insertion into a table

Physical Data Modelling

Inserting Data Into A Table:

Several methods can be used to enter data into a table.

The INSERT Statement:

The INSET INTO statement is used to insert or add a row(s) of data into a table. The VALUE clause specifies the actual data of the row that will be added to the table. These statements can be specified using several methods.

METHOD 1: Specifying Column Names With Values

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Physical Data Modelling

Insert data for customer 912637 into the CUSTOMERS table using an explicit column list.

```
INSERT INTO customers
  ( customer_id, customer_name, balance, ship_city, credit_limit, discount )
VALUES ( 912637, 'Brideview Inc.', 0.00, 'Dallas', 50000, .050 );
```

Results:

Succeeded.

```
SELECT *
FROM customers
WHERE customer_id = 912637;
```

Results:

CUSTOMER_ID	CUSTOMER_NAME	BALANCE	SHIP_CITY	CREDIT_LIMIT	DISCOUNT
912637	Brideview Inc.	0.00	Dallas	50000	0.050

Physical Data Modelling

METHOD 2: Specifying Values Without Column Names

To insert a row into table without using column names, enter the keywords INSERT INTO followed by the name of the table into which a row of data is being added. The VALUES command follows the INSERT INTO command and specifies the values of the columns as shown below

```
INSERT INTO employees
VALUES ( 02,
        7315,
        'Amy',
        'Q',
        'Programmer',
        '2015-05-10',
        666,
        25.00,
        40.5,
        55555 );
```

Why this error ?



```
INSERT INTO employees
VALUES ( 03,
        7315,
        'Amy',
        'Q',
        'Programmer',
        666,
        25.00,
        40.5,
        55555 );
```

ERROR MESSAGE

Physical Data Modelling

METHOD 3: Specifying Multiple Values With one INSERT statement

Several rows can be inserted into a table at one time. As shown in the following figure four employees are being inserted in the EMPLOYEES table using one INSERT statement

```
INSERT INTO employees VALUES
(1, 'Angela', 'Bailey', 'F', '2001-11-11'),
(2, 'Nancy', 'Lopez', 'F', '2002-05-20'),
(3, 'Stephen', 'Reynolds', 'M', '2003-03-21'),
(4, 'Benjamin', 'Adams', 'M', '2003-03-22');
```

Inserting Rows with NULL Values

A column that is set to null must have been defined without the NOT NULL clause in the CREATE TABLE command.

The absence of NOT NULL indicates that this column is null-capable. If a column can hold null values, it can be omitted from the INSERT statement. An implicit insert will automatically insert a null value in that column

Physical Data Modelling

```
-- INSERT 1
INSERT INTO customers
  ( customer_id, customer_name, balance, ship_city, credit_limit )
VALUES ( 954934, 'Crane Inc.', 0, 'Detroit', 75000 );

-- INSERT 2
INSERT INTO customers
  ( customer_id, customer_name, balance, ship_city, credit_limit, discount )
VALUES ( 969124, 'KTS Consulting', 0, 'Dallas', 125000, DEFAULT );

-- INSERT 3
INSERT INTO customers
  ( customer_id, customer_name, balance, ship_city, credit_limit, discount )
VALUES ( 972753, 'RJ Young Company', 0, 'Houston', 85000, NULL );
```

Insert three new customers into the CUSTOMERS table. Use three methods to insert a NULL value into the discount column.

Results:

Succeeded.
Succeeded.
Succeeded.

Physical Data Modelling

Inserting Rows with DEFAULT Values

When a default value is specified for a column, the same format as inserting a NULL value is used. When a column name is not specified on the INSERT INTO statement and it has been defined with a default value, default value is automatically inserted into the column

A column in a table can be given a default value. This option prevents null values from entering the columns if a row inserted without a specified value for the column

Insert a new customer into the CUSTOMERS table and specify the DEFAULT value for the balance and credit_limit columns.

```
INSERT INTO customers
(customer_id, customer_name, balance, ship_city, credit_limit, discount)
VALUES ( 901758, 'ATD Inc.', DEFAULT, 'Jacksonville', DEFAULT, .060 );
```


Physical Data Modelling

INSERT Statement with DEFAULT Date

There are many instances where a default date is required. For example , all orders placed on a particular day might be inserted using the current date

Insert employee 139 into the `EMPLOYEES_RETIRED` table. Use `CURRENT_DATE` for the `retire_date` column.

```
INSERT INTO employees_retired
( employee_id, first_name, middle_initial, last_name, retire_date )
VALUES ( 139, 'Rick', 'D', 'Peters', CURRENT_DATE );
```

Physical Data Modelling

Copy Data from One Table to Another

An INSERT statement can be used to copy data from one table to another table. Following INSEET statement copies all rows from the EMPLOYEES_RETIRED table to the EMPLOYEES_RETIRED_COPY table.

In addition , the statement includes an additional column, last_job_code , in the EMPLOYEES_RETIRED_COPY table

```
INSERT INTO employees_retired_copy ( employee_id, first_name, middle_initial,  
                                     last_name, retire_date, last_job_code )  
  SELECT employee_id, first_name, middle_initial, last_name, retire_date, ''  
 FROM employees_retired;
```

Physical Data Modelling

The UPDATE Statement

The UPDATE statement modifies existing rows in a table. It requires four values

The Name of the table.

The name of at least one column to modify

A value for each column(s) being modified

The optional WHERE clause that identifies the row or rows in the table to be modified.

```
UPDATE customers
SET customer_name = 'KTS Consulting Inc.',
    discount = discount + .02
WHERE customer_id = 969124;
```

What will happen if we omit
WHERE Clause ???

Physical Data Modelling

DELETE Statement

The DELETE statement removes one or more existing rows from a table. The Statement requires two values

The Name of the table

The condition that identifies the rows to be deleted

```
DELETE  
FROM customers  
WHERE customer_id = 905011;
```

Delete (clear) all rows in the CUSTOMERS table.

```
DELETE  
FROM customers;
```

Deleting rows from multiple tables require multiple DELETE statements to be executed. A foreign key constraint can be used with a **DELETE CASCADE** rule to delete all dependent rows along with a parent row.

Conclusion

