Database Programming with SQL
12-1: INSERT Statements

Objectives

- Give examples of why it is important to be able to alter the data in a database
- Construct and execute INSERT statements that insert a single row using a VALUES clause
- Construct and execute INSERT statements that use special values, null values, and date values
- Construct and execute INSERT statements that copy rows from one table to another using a subquery

Vocabulary

Identify the vocabulary word for each definition below.

a. Someone doing "real work" with the computer, using it as a means rather than an end END USER
b. Consists of a collection of DML statements that form a logical unit of work. TRANSACTION
c. Fully and clearly expressed; leaving nothing implied EXPLICIT
d. Adds a new row to a table INSERT

Try It / Solve It

Students should execute DESC tablename before doing INSERT to view the data types for each column. VARCHAR2 data-type entries need single quotation marks in the VALUES statement.

1. Give two examples of why it is important to be able to alter the data in a database.
   - Alter data in a database to keep information current. Example, updating a patient's contact or insurance details ensures accurate records for billing and care decisions.
   - Altering data allows for correcting mistakes: misspelled names or incorrect inventory counts, which helps maintain the accuracy and reliability of the database for users and decision-makers.

2. DJs on Demand just purchased four new CDs. Use an explicit INSERT statement to add each CD to the copy_d_cds table. After completing the entries, execute a SELECT * statement to verify your work.

CD_Number	Title	Producer	Year
97	Celebrate the Day	R & B Inc.	2003
98	Holiday Tunes for All Ages	Tunes are Us	2004
99	Party Music	Old Town Records	2004
100	Best of Rock and Roll	Old Town Records	2004

INSERT INTO copy_d_cds (cd_id, cd_title, artist_name, release_year, genre)
VALUES (1, 'Party Music', 'Holiday Tunes for All ages', 'Best of Rock and Roll', 2004, 'Celebrate the Day'');
SELECT * FROM copy_d_cds;

3. DJs on Demand has two new events coming up. One event is a fall football party and the other event is a sixties theme party. The DJs on Demand clients requested the songs shown in the table for their events. Add these songs to the copy_d_songs table using an implicit INSERT statement.

ID	Title	Duration	Type_Code
52	Surfing Summer	Not known	12
53	Victory Victory	5 min	12

INSERT INTO copy_d_songs
VALUES (1, 'Surfing Summer', 'Victory Victory', 12);
SELECT * FROM copy_d_songs;

4. Add the two new clients to the copy_d_clients table. Use either an implicit or an explicit INSERT.

Client_Number	First_Name	Last_Name	Phone	Email
6655	Ayako	Dahish	3608859030	dahisha@harbor.net
6689	Nick	Neuville	9048953049	nnicky@charter.net

INSERT INTO copy_d_clients (client_id, client_name, client_contact, client_address)
VALUES (1, 'Ayako Dahish', 'dahisha@harbor.net', '3608859030');
SELECT * FROM copy_d_clients;

5. Add the new client's events to the copy_d_events table. The cost of each event has not been determined at this date.

ID	Name	Event_ Date	Description	Cost	Venue_ ID	Package_ Code	Theme_ Code	Client_ Number
110	Ayako Anniversary	07-Jul-2004	Party for 50, sixties dress, decorations		245	79	240	6655
115	Neuville Sports Banquet	09-Sep-2004	Barbecue at residence, college alumni, 100 people		315	87	340	6689

INSERT INTO copy_d_events (event_id, event_name, client_id, event_date)
VALUES (1, 'Party for 50, sixties dress, decorations', 110, '07-Jul-2004');
SELECT * FROM copy_d_events;

6. Create a table called rep_email using the following statement:
CREATE TABLE rep_email (
id NUMBER(3) CONSTRAINT rel_id_pk PRIMARY KEY,
first_name VARCHAR2(10),
last_name VARCHAR2(10),
email_address VARCHAR2(10))
Populate this table by running a query on the employees table that includes only those employees who are REP's.

INSERT INTO email (id, first_name, last_name, email_address)
SELECT employee_id, first_name, last_name, SUBSTR(email, 1, 10) VARCHAR2(10)
FROM employees
WHERE job_title = 'REP';

_

Database Programming with SQL
12-2: Updating Column Values and Deleting Rows

Objectives

- Construct and execute an UPDATE statement
- Construct and execute a DELETE statement
- Construct and execute a query that uses a subquery to update and delete data from a table
- Construct and execute a query that uses a correlated subquery to update and delete from a table
- Explain how foreign-key and primary-key integrity constraints affect UPDATE and DELETE statements
- Explain the purpose of the FOR UPDATE Clause in a SELECT statement

Vocabulary

Identify the vocabulary word for each definition below.

a. Modifies existing rows in a table retrieves information from one table & uses the information to update another table
   i. *UPDATE*
b. Ensures that the data adheres to a predefined set of rules deletes information on a linked table based on what was deleted on the other table
   i. *FOREIGN KEY CONSTRAINT*
c. Removes existing rows from a table
   i. *DELETE*

Try It / Solve It

NOTE: Copy tables in this section do not exist

If any change is not possible, give an explanation as to why it is not possible.

1. Monique Tuttle, the manager of Global Fast Foods, sent a memo requesting an immediate change in prices. The price for a strawberry shake will be raised from $3.59 to $3.75, and the price for fries will increase to $1.20. Make these changes to the copy_f_food_items table.

```
UPDATE copy_f_food_items
SET price = 3.75
WHERE item_name = 'Strawberry Shake' AND price = 3.59;

UPDATE copy_f_food_items
SET price = 1.20
WHERE item_name = 'Fries';
```

2. Bob Miller and Sue Doe have been outstanding employees at Global Fast Foods. Management has decided to reward them by increasing their overtime pay. Bob Miller will receive an additional $0.75 per hour and Sue Doe will receive an additional $0.85 per hour. Update the copy_f_staffs table to show these new values. (Note: Bob Miller currently doesn't get overtime pay. What function do you need to use to convert a null value to 0?)

```
UPDATE copy_f_staffs
SET overtime_pay = NVL(overtime_pay, 0) + 0.75
WHERE first_name = 'Bob' AND last_name = 'Miller';

UPDATE copy_f_staffs
SET overtime_pay = overtime_pay + 0.85
WHERE first_name = 'Sue' AND last_name = 'Doe';
```

3. Add the orders shown to the Global Fast Foods copy_f_orders table:

ORDER_NUMBER	ORDER_DATE	ORDER_TOTAL	CUST_ID	STAFF_ID
5680	June 12, 2004	159.78	145	9
5691	09-23-2004	145.98	225	12
5701	July 4, 2004	229.31	230	12
5702	10-31-2004	3.59	235	12
5703	11-20-2004	1.20	240	12

4. Add the new customers shown below to the copy_f_customers table. You may already have added Katie Hernandez. Will you be able to add all these records successfully?

ID	FIRST_ NAME	LAST_ NAME	ADDRESS	CITY	STATE	ZIP	PHONE_NUMBER
145	Katie	Hernandez	92 Chico Way	Los Angeles	CA	98008	8586667641
225	Daniel	Spode	1923 Silverado	Denver	CO	80219	7193343523
230	Adam	Zurn	5 Admiral Way	Seattle	WA		4258879009

5. Sue Doe has been an outstanding Global Foods staff member and has been given a salary raise. She will now be paid the same as Bob Miller. Update her record in copy_f_staffs.

```
UPDATE copy_f_staffs SET salary = (SELECT salary FROM copy_f_staffs WHERE first_name = 'Bob' AND last_name = 'Miller')
WHERE first_name = 'Sue' AND last_name = 'Doe';
```

6. Global Fast Foods is expanding their staff. The manager, Monique Tuttle, has hired Kai Kim. Not all information is available at this time, but add the information shown here.

ID	FIRST_NAME	LAST_NAME	BIRTHDATE	SALARY	STAFF_TYPE
25	Kai	Kim	3-Nov-1988	6.75	Order Taker

7. Now that all the information is available for Kai Kim, update his Global Fast Foods record to include the following: Kai will have the same manager as Sue Doe. He does not qualify for overtime. Leave the values for training, manager budget, and manager target as null.

```
UPDATE copy_f_staffs
SET manager_id = (SELECT manager_id FROM copy_f_staffs WHERE first_name = 'Sue' AND last_name = 'Doe'), overtime_pay = NULL,    -- Kai does not qualify for overtime training = NULL, manager_budget = NULL,  -- Leave manager budget as NULL manager_target = NULL
```

WHERE first_name = 'Kai' AND last_name = 'Kim';

8. Execute the following SQL statement. Record your results.
DELETE from departments
WHERE department_id = 60;
SELECT * FROM departments
WHERE department_id = 60;
9. Kim Kai has decided to go back to college and does not have the time to work and go to school. Delete him from the Global Fast Foods staff. Verify that the change was made.
DELETE FROM copy_f_staffs
WHERE first_name = 'Kai' AND last_name = 'Kim';
10. Create a copy of the employees table and call it lesson7_emp;
Once this table exists, write a correlated delete statement that will delete any employees from the lesson7_employees table that also exist in the job_history table
DELETE FROM lesson7_emp le
WHERE EXISTS (SELECT 1 FROM job_history jh WHERE le.employee_id = jh.employee_id);

Database Programming with SQL
12-3: DEFAULT Values, MERGE, and Multi-Table Inserts

Objectives

- Understand when to specify a DEFAULT value
- Construct and execute a MERGE statement
- Construct and execute DML statements using SUBQUERIES
- Construct and execute multi-table inserts

Try It / Solve It

1. When would you want a DEFAULT value?
When there is consistency in data, when handling optional data, make sure of data integrity, and to reduce errors
2. Currently, the Global Foods F_PROMOTIONAL_MENUS table START_DATE column does not have SYSDATE set as DEFAULT. Your manager has decided she would like to be able to set the starting date of promotions to the current day for some entries. This will require three steps:

a. In your schema, Make a copy of the Global Foods F_PROMOTIONAL_MENUS table using the following SQL statement:
CREATE TABLE copy_f_promotional_menus
AS (SELECT * FROM f_promotional_menus)

b. Alter the current START_DATE column attributes using:
ALTER TABLE copy_f_promotional_menus
MODIFY(start_date DATE DEFAULT SYSDATE)

c. INSERT the new information and check to verify the results.
INSERT a new row into the copy_f_promotional_menus table for the manager's new promotion. The promotion code is 120. The name of the promotion is 'New Customer.' Enter DEFAULT for the start date and '01-Jun-2005' for the ending date. The giveaway is a 10% discount coupon. What was the correct syntax used?
INSERT INTO copy_f_promotional_menus (promotion_code, promotion_name, start_date, end_date, giveaway)
VALUES (120, 'New Customer', DEFAULT, TO_DATE('01-JUN-2005', 'DD-MON-YYYY'), '10% discount coupon');
3. Allison Plumb, the event planning manager for DJs on Demand, has just given you the following list of CDs she acquired from a company going out of business. She wants a new updated list of CDs in inventory in an hour, but she doesn't want the original D_CDS table changed. Prepare an updated inventory list just for her.

a. Assign new cd_numbers to each new CD acquired.

b. Create a copy of the D_CDS table called manager_copy_d_cds. What was the correct syntax used?
CREATE TABLE manager_copy_d_cds AS
SELECT * FROM D_CDS;
c. INSERT into the manager_copy_d_cds table each new CD title using an INSERT statement. Make up one example or use this data:
20, 'Hello World Here I Am', 'Middle Earth Records', '1998'
What was the correct syntax used?
INSERT INTO manager_copy_d_cds (cd_number, title, record_label, release_year)
VALUES (20, 'Hello World Here I Am', 'Middle Earth Records', '1998');
d. Use a merge statement to add to the manager_copy_d_cds table, the CDs from the original table. If there is a match, update the title and year. If not, insert the data from the original table. What was the correct syntax used?
MERGE INTO manager_copy_d_cds m
USING D_CDS ON (cd_number = cd_number)
WHEN MATCHED THEN UPDATE SET title = title, release_year = release_year
WHEN NOT MATCHED THEN INSERT (cd_number, title, record_label, release_year),
VALUES (d.cd_number, d.title, d.record_label, d.release_year);

4. Run the following 3 statements to create 3 new tables for use in a Multi-table insert statement. All 3 tables should be empty on creation, hence the WHERE 1=2 condition in the WHERE clause.

```
CREATE TABLE sal_history (employee_id, hire_date, salary)
AS SELECT employee_id, hire_date, salary
FROM employees
WHERE 1=2;

CREATE TABLE mgr_history (employee_id, manager_id, salary)
AS SELECT employee_id, manager_id, salary
FROM employees
WHERE 1=2;

CREATE TABLE special_sal (employee_id, salary)
AS SELECT employee_id, salary
FROM employees
WHERE 1=2;
```

Once the tables exist in your account, write a Multi-Table insert statement to first select the employee_id, hire_date, salary, and manager_id of all employees. If the salary is more than 20000 insert the employee_id and salary into the special_sal table. Insert the details of employee_id, hire_date, and salary into the sal_history table. Insert the employee_id, manager_id, and salary into the mgr_history table.

```
INSERT ALL
   INTO special_sal (employee_id, salary)
      SELECT employee_id, salary
      FROM employees
      WHERE salary > 20000
   INTO sal_history (employee_id, hire_date, salary)
      SELECT employee_id, hire_date, salary
      FROM employees
   INTO mgr_history (employee_id, manager_id, salary)
      SELECT employee_id, manager_id, salary
      FROM employees
SELECT * FROM dual;
```

You should get a message back saying 39 rows were inserted. Verify you get this message and verify you have the following number of rows in each table:

Sal_history: 19 rows
Mgr_history: 19 rows
Special_sal: 1

_

Database Programming with SQL
13-1: Creating Tables

Objectives

- List and categorize the main database objects
- Review a table structure
- Describe how database schema objects are used by the Oracle database

Vocabulary

Identify the vocabulary word for each definition below.

a. Created and maintained by the Oracle Server and contains information about the database Data Dictionary
b. A collection of objects that are the logical structures that directly refer to the data in the database Schema
c. Specifies a preset value if a value is omitted in the INSERT statement DEFAULT
d. Stores data; basic unit of storage composed of rows and columns table
e. Command use to make a new table CREATE TABLE

Try It / Solve It

1. Complete the GRADUATE CANDIDATE table instance chart. Credits is a foreign-key column referencing the requirements table.

Column Name	student_id	last_name	first_name	credits	graduation_date
Key Type					
Nulls/Unique					
FK Column					
Datatype	NUMBER	VARCHAR2	VARCHAR2	NUMBER	DATE
Length	6			3	

2. Write the syntax to create the grad_candidates table.
CREATE TABLE grad_candidates (candidate_id NUMBER(10) PRIMARY KEY, first_name VARCHAR2(50) NOT NULL, last_name VARCHAR2(50) NOT NULL, email VARCHAR2(100),
3. Confirm creation of the table using DESCRIBE.
DESCRIBE grad_candidates;

4. Create a new table using a subquery. Name the new table your last name -- e.g., smith_table. Using a subquery, copy grad_candidates into smith_table.

```
CREATE TABLE smith_table AS
SELECT *
FROM grad_candidates;
```

5. Insert your personal data into the table created in question 4.

```
INSERT INTO lastname_table (candidate_id, first_name, last_name, email, phone_number,
graduation_date, degree_program, gpa, status)
VALUES (1, 'FirstName', 'LastName', 'email@example.com', '1234567890',
TO_DATE('15-Nov-2024', 'DD-Mon-YYYY'));
```

6. Query the data dictionary for each of the following:

- USER_TABLES
- USER_OBJECTS
- USER_CATALOG or USER_CAT

In separate sentences, summarize what each query will return

_

Database Programming with SQL
13-2: Using Data Types

Objectives

- Create a table using TIMESTAMP and TIMESTAMP WITH TIME ZONE column data types
- Create a table using INTERVAL YEAR TO MONTH and INTERVAL DAY TO SECOND column data types
- Give examples of organizations and personal situations where it is important to know to which time zone a date-time value refers
- List and provide an example of each of the number, date, and character data types

Vocabulary

Identify the vocabulary word for each definition below.

a. Allows time to be stored as an interval of years and months INTERVAL
b. When a column is selected in a SQL statement the time is automatically converted to the user's timezone TIMESTAMP
c. Binary large object data up to 4 gigabytes BLOB
d. Stores a time zone value as a displacement from Universal TIMESTAMP
e. Coordinated Time or UCT INTERVAL DAY TO SECOND

f. Allows time to be stored as an interval of days to hours, minutes, and seconds CLOB
g. Character data up to 4 gigabytes CLOB
h. Allows the time to be stored as a date with fractional seconds TIMESTAMP

Try It / Solve It

1. Create tables using each of the listed time-zone data types, use your time-zone and one other in your examples. Answers will vary.

a. TIMESTAMP WITH LOCAL TIME ZONE
CREATE TABLE timestamp_local_tz_example (id NUMBER PRIMARY KEY, event_name VARCHAR2(100), event_timestamp TIMESTAMP WITH LOCAL TIME ZONE);
b. INTERVAL YEAR TO MONTH
CREATE TABLE interval_year_month_example (id NUMBER PRIMARY KEY, project_name VARCHAR2(100), project_duration INTERVAL YEAR TO MONTH);
c. INTERVAL DAY TO SECOND
CREATE TABLE interval_day_second_example (id NUMBER PRIMARY KEY, task_name VARCHAR2(100), task_duration INTERVAL DAY TO SECOND);
2. Execute a SELECT * from each table to verify your input.
SELECT *
FROM timestamp_loca;
3. Give 3 examples of organizations and personal situations where it is important to know to which time zone a date-time value refers
Scheduling, telemedicine, ordering process

_

Database Programming with SQL
13-3: Modifying a Table

Objectives

- Explain why it is important to be able to modify a table
- Explain and provide an example for each of the DDL statements—ALTER, DROP, RENAME, and TRUNCATE—and the effect each has on tables and columns
- Construct a query and execute the ALTER TABLE commands ADD, MODIFY, and DROP
- Explain and perform a FLASHBACK QUERY on a table
- Explain and perform FLASHBACK table operations
- Track the changes to data over a period of time
- Explain the rationale for using TRUNCATE versus DELETE for tables

- Add a comment to a table using the COMMENT ON TABLE command
- Name the changes that can and cannot be made to modify a column
- Explain when and why the SET UNUSED statement is advantageous

Try It / Solve It

Before beginning the practice exercises, execute a DESCRIBE for each of the following tables: o_employees, o_departments and o_jobs. These tables will be used in the exercises. If they do not exist in your account, create them as follows:

1. Create the three o_tables – jobs, employees, and departments – using the syntax:

```
CREATE TABLE o_jobs AS (SELECT * FROM jobs);
CREATE TABLE o_employees AS (SELECT * FROM employees);
CREATE TABLE o_departments AS (SELECT * FROM departments);
```

Job	Employee ID	Department
Data analyst	23245	Lab
Project Manager	32435	Lab