

12-1: INSERT Statements

Vocabulary

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

Try It / Solve It

Give two examples of why it is important to be able to alter the data in a database

- Keeping information current
- Correcting information / updating information

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

- 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;

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.

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

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

- 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;

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

- j. 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;

Create a table called rep_email using the following statement:

- k. CREATE TABLE rep_email (
- l. id NUMBER(3) CONSTRAINT rel_id_pk PRIMARY KEY,
- m. first_name VARCHAR2(10),
- n. last_name VARCHAR2(10),
- o. email_address VARCHAR2(100))
- p. Populate this table by running a query on the employees table that includes only those employees who are REP's.
 - i. INSERT INTO email (id, first_name, last_name, email_address)
 - ii. SELECT employee_id, first_name, last_name, SUBSTR(email, 1, 10)
VARCHAR2(10) FROM employees
 - iii. WHERE job_title = 'REP';

12-2: Updating Column Values and Deleting Rows

| | |
|------------------------|---|
| UPDATE | Modifies existing rows in a table retrieves information from one table & uses the information to update another table |
| FOREIGN KEY CONSTRAINT | 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 |
| DELETE | Removes existing rows from a table |

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
 - a. UPDATE copy_f_food_items
 - b. SET price = 3.75
 - c. WHERE item_name = 'Strawberry Shake' AND price = 3.59;
 - d. UPDATE copy_f_food_items
 - e. SET price = 1.20
 - f. 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?)

- a. UPDATE copy_f_staffs
 - b. SET overtime_pay = NVL(overtime_pay, 0) + 0.75
 - c. WHERE first_name = 'Bob' AND last_name = 'Miller';
 - d. UPDATE copy_f_staffs
 - e. SET overtime_pay = overtime_pay + 0.85
 - f. WHERE first_name = 'Sue' AND last_name = 'Doe';
3. Add the orders shown to the Global Fast Foods copy_f_orders table:
 - a.

| | | |
|------|------------|-------------|
| 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?
 - a.
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.
 - a. UPDATE copy_f_staffs SET salary = (SELECT salary FROM copy_f_staffs WHERE first_name = 'Bob' AND last_name = 'Miller')
 - b. 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.
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.
 - a. UPDATE copy_f_staffs
 - b. 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
 - c. WHERE first_name = 'Kai' AND last_name = 'Kim';
8. Execute the following SQL statement. Record your results. DELETE from departments. WHERE department_id = 60;
 - a. DELETE FROM copy_f_staffs
 - b. WHERE first_name = 'Kai' AND last_name = 'Kim';
9. 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
 - a. DELETE FROM lesson7_emp le

- b. WHERE EXISTS (SELECT 1 FROM job_history jh WHERE le.employee_id = jh.employee_id);

12-3: DEFAULT Values, MERGE, and Multi-Table Inserts

1. When would you want a DEFAULT value?
 - a. 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:
 - b. CREATE TABLE copy_f_promotional_menus
 - c. AS (SELECT * FROM f_promotional_menus)
 - d. Alter the current START_DATE column attributes using:
 - e. ALTER TABLE copy_f_promotional_menus
 - f. MODIFY(start_date DATE DEFAULT SYSDATE)
 - g. INSERT the new information and check to verify the results.
 - h. 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?
 - i. INSERT INTO copy_f_promotional_menus (promotion_code, promotion_name, start_date, end_date, giveaway)
 - j. 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?
 - i. CREATE TABLE manager_copy_d_cds AS
 - ii. SELECT * FROM D_CDS;
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.
 - a. CREATE TABLE sal_history (employee_id, hire_date, salary)
 - b. AS SELECT employee_id, hire_date, salary

- c. FROM employees
- d. WHERE 1=2;
 - i. CREATE TABLE mgr_history (employee_id, manager_id, salary)
 - ii. AS SELECT employee_id, manager_id, salary
 - iii. FROM employees
 - iv. WHERE 1=2;
- e. CREATE TABLE special_sal (employee_id, salary)
- f. AS SELECT employee_id, salary
- g. FROM employees
- h. WHERE 1=2;

13-1 Creating Tables

1. Complete the GRADUATE CANDIDATE table instance chart. Credits is a foreign-key column referencing the requirements table.
2. Write the syntax to create the grad_candidates table.
 - a. 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.
 - a. 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
 - a. CREATE TABLE smith_table AS
 - b. SELECT *
 - c. FROM grad_candidates;