Programming 12, 13

Katherine Hackett

12-1

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 statement - Adds a new row to a table

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.

* **Keeping Information Current**: Businesses need to keep records up-to-date, such as updating customer contact information, inventory levels, or product prices to ensure accuracy in decision-making.
* **Correcting Errors**: Users may need to correct data entry mistakes, such as an incorrect date or misspelled name, to maintain data integrity and prevent issues in reporting or analysis.

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.

INSERT INTO copy\_d\_cds (CD\_Number, Title, Producer, Year)

VALUES (97, 'Celebrate the Day', 'R & B Inc.', 2003);

INSERT INTO copy\_d\_cds (CD\_Number, Title, Producer, Year)

VALUES (98, 'Holiday Tunes for All Ages', 'Tunes are Us', 2004);

INSERT INTO copy\_d\_cds (CD\_Number, Title, Producer, Year)

VALUES (99, 'Party Music', 'Old Town Records', 2004);

INSERT INTO copy\_d\_cds (CD\_Number, Title, Producer, Year)

VALUES (100, 'Best of Rock and Roll', 'Old Town Records', 2004);

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.

INSERT INTO copy\_d\_songs

VALUES (52, 'Surfing Summer', 'Not known', 12);

INSERT INTO copy\_d\_songs

VALUES (53, 'Victory Victory', '5 min', 12);

4. 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\_Number, First\_Name, Last\_Name, Phone, Email)

VALUES (6655, 'Ayako', 'Dahish', '3608859030', 'dahisha@harbor.net');

INSERT INTO copy\_d\_clients (Client\_Number, First\_Name, Last\_Name, Phone, Email)

VALUES (6689, 'Nick', 'Neuville', '9048953049', 'nnicky@charter.net');

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.

INSERT INTO copy\_d\_events (ID, Name, Event\_Date, Description, Cost, Venue\_ID, Package\_Code, Theme\_Code, Client\_Number)

VALUES (110, 'Ayako Anniversary', TO\_DATE('07-Jul-2004', 'DD-Mon-YYYY'), 'Party for 50, sixties dress, decorations', NULL, 245, 79, 240, 6655);

INSERT INTO copy\_d\_events (ID, Name, Event\_Date, Description, Cost, Venue\_ID, Package\_Code, Theme\_Code, Client\_Number)

VALUES (115, 'Neuville Sports Banquet', TO\_DATE('09-Sep-2004', 'DD-Mon-YYYY'), 'Barbecue at residence, college alumni, 100 people', NULL, 315, 87, 340, 6689);

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.

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)

);

INSERT INTO rep\_email (id, first\_name, last\_name, email\_address)

SELECT id, first\_name, last\_name, email\_address

FROM employees

WHERE job\_title = 'REP';

12-2

Vocabulary

UPDATE statement - Modifies existing rows in a table

Correlated Subquery - Retrieves information from one table & uses the information to update another table

Constraints - Ensures that the data adheres to a predefined set of rules

On Delete Cascade - Deletes information on a linked table based on what was deleted on the other table

Delete statement - 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.

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 = NVL(overtime\_pay, 0) + 0.85

WHERE first\_name = 'Sue' AND last\_name = 'Doe';

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 = NVL(overtime\_pay, 0) + 0.85

WHERE first\_name = 'Sue' AND last\_name = 'Doe';

3. Add the orders shown to the Global Fast Foods copy\_f\_orders table:

INSERT INTO copy\_f\_orders (ORDER\_NUMBER, ORDER\_DATE, ORDER\_TOTAL, CUST\_ID, STAFF\_ID)

VALUES (5680, TO\_DATE('12-Jun-2004', 'DD-Mon-YYYY'), 159.78, 145, 9);

INSERT INTO copy\_f\_orders (ORDER\_NUMBER, ORDER\_DATE, ORDER\_TOTAL, CUST\_ID, STAFF\_ID)

VALUES (5691, TO\_DATE('23-Sep-2004', 'DD-Mon-YYYY'), 145.98, 225, 12);

INSERT INTO copy\_f\_orders (ORDER\_NUMBER, ORDER\_DATE, ORDER\_TOTAL, CUST\_ID, STAFF\_ID)

VALUES (5701, TO\_DATE('04-Jul-2004', 'DD-Mon-YYYY'), 229.31, 230, 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?

INSERT INTO copy\_f\_customers (ID, FIRST\_NAME, LAST\_NAME, ADDRESS, CITY, STATE, ZIP, PHONE\_NUMBER)

VALUES (145, 'Katie', 'Hernandez', '92 Chico Way', 'Los Angeles', 'CA', '98008', '8586667641');

INSERT INTO copy\_f\_customers (ID, FIRST\_NAME, LAST\_NAME, ADDRESS, CITY, STATE, ZIP, PHONE\_NUMBER)

VALUES (225, 'Daniel', 'Spode', '1923 Silverado', 'Denver', 'CO', '80219', '7193343523');

INSERT INTO copy\_f\_customers (ID, FIRST\_NAME, LAST\_NAME, ADDRESS, CITY, STATE, ZIP, PHONE\_NUMBER)

VALUES (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.

INSERT INTO copy\_f\_staffs (ID, FIRST\_NAME, LAST\_NAME, BIRTHDATE, SALARY, STAFF\_TYPE)

VALUES (25, 'Kai', 'Kim', TO\_DATE('03-Nov-1988', 'DD-Mon-YYYY'), 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 = 0,

training = NULL,

manager\_budget = 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;

DELETE 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';

-- Verify deletion

SELECT \* 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.

CREATE TABLE lesson7\_emp AS

SELECT \* FROM employees;

DELETE FROM lesson7\_emp

WHERE id IN (SELECT employee\_id FROM job\_history);

12-3

1. When would you want a DEFAULT value?

* You want a column to have a common initial value if no specific value is provided during data entry.
* You’re tracking dates, timestamps, or statuses, and you want new rows to default to the current date or an initial status.
* You want to ensure data consistency without requiring every user to specify a value.

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, 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, producer, 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 mc

USING D\_CDS d

ON (mc.cd\_number = d.cd\_number)

WHEN MATCHED THEN

UPDATE SET mc.title = d.title, mc.year = d.year

WHEN NOT MATCHED THEN

INSERT (cd\_number, title, producer, year)

VALUES (d.cd\_number, d.title, d.producer, d.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.

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

13-1

Vocabulary

Data Dictionary - Created and maintained by the Oracle Server and contains information about the database

Schema - A collection of objects that are the logical structures that directly refer to the data in the database

Default Constraint - Specifies a preset value if a value is omitted in the INSERT statement

Table - Stores data; basic unit of storage composed of rows and columns

Create Table Statement - Command use to make a new table

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

| **Column Name** | **Key Type** | **Nulls/Unique** | **FK Column** | **Datatype** | **Length** |
| --- | --- | --- | --- | --- | --- |
| student\_id | Primary | Not Null/Unique | No | NUMBER | 6 |
| last\_name |  | Not Null | No | VARCHAR2 | 50 |
| first\_name |  | Not Null | No | VARCHAR2 | 50 |
| credits |  | Nullable | Yes | NUMBER | 3 |
| graduation\_date |  | Nullable | No | DATE |  |

2. Write the syntax to create the grad\_candidates table.

CREATE TABLE grad\_candidates (

student\_id NUMBER(6) PRIMARY KEY,

last\_name VARCHAR2(50) NOT NULL,

first\_name VARCHAR2(50) NOT NULL,

credits NUMBER(3),

graduation\_date DATE,

CONSTRAINT fk\_credits FOREIGN KEY (credits) REFERENCES requirements(credits)

);

3. Confirm creation of the table using DESCRIBE.

DESC 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.

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.

SELECT \* FROM USER\_OBJECTS; etc.

13-2

Interval year to month - Allows time to be stored as an interval of years and months

Timestamp with timezone - When a column is selected in a SQL statement the time is automatically converted to the user’s timezone

BLOB (Binary large object) - Binary large object data up to 4 gigabytes

Timestamp with time zone - Stores a time zone value as a displacement from Universal Coordinated Time or UCT

Interval day to second - Allows time to be stored as an interval of days to hours, minutes, and seconds

CLOB (Character large object) - Character data up to 4 gigabytes

Timestamp - Allows the time to be stored as a date with fractional seconds

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\_timezone\_example (

id NUMBER PRIMARY KEY,

event\_name VARCHAR2(100),

event\_time TIMESTAMP WITH LOCAL TIME ZONE

);

INSERT INTO timestamp\_local\_timezone\_example (id, event\_name, event\_time)

VALUES (1, 'Conference Call', TO\_TIMESTAMP\_TZ('2024-11-12 14:00:00 America/New\_York', 'YYYY-MM-DD HH24:MI:SS TZD'));

b. INTERVAL YEAR TO MONTH

CREATE TABLE interval\_year\_month\_example (

id NUMBER PRIMARY KEY,

event\_name VARCHAR2(100),

duration INTERVAL YEAR TO MONTH

);

INSERT INTO interval\_year\_month\_example (id, event\_name, duration)

VALUES (1, 'Membership Duration', INTERVAL '2' YEAR);

c. INTERVAL DAY TO SECOND

CREATE TABLE interval\_day\_second\_example (

id NUMBER PRIMARY KEY,

event\_name VARCHAR2(100),

duration INTERVAL DAY TO SECOND

);

INSERT INTO interval\_day\_second\_example (id, event\_name, duration)

VALUES (1, 'Flight Duration', INTERVAL '3 14:30:00' DAY TO SECOND);

2. Execute a SELECT \* from each table

to verify your input.

SELECT \* FROM timestamp\_local\_timezone\_example;

SELECT \* FROM interval\_year\_month\_example;

SELECT \* FROM interval\_day\_second\_example;

3. Give 3 examples of organizations and personal situations where it is important to know to which time zone a date-time value refers.

Amazon delivery estimations, Delta Airlines, Zoom meetings

13-3

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);

2. Add the Human Resources job to the jobs table:

INSERT INTO o\_jobs (job\_id, job\_title, min\_salary, max\_salary)

VALUES('HR\_MAN', 'Human Resources Manager', 4500, 5500);

3. Add the three new employees to the employees table:

INSERT INTO o\_employees (employee\_id, first\_name, last\_name, email, hire\_date, job\_id)

VALUES(210, 'Ramon', 'Sanchez', 'RSANCHEZ', SYSDATE, 'HR\_MAN');

4. Add Human Resources to the departments table:

INSERT INTO o\_departments(department\_id, department\_name)

VALUES (210,'Human Resources');

You will need to know which columns do not allow null values.

1. Why is it important to be able to modify a table?

**Correcting Errors**: Data might be incorrect or incomplete, and modifications (updates or additions) allow fixing or enriching the information.

**Adapting to New Requirements**: As business requirements evolve, the database schema may need to change (e.g., adding columns, changing data types, etc.) to accommodate new features or data.

**Efficiency and Maintenance**: Modifying a table (e.g., adding indexes or altering constraints) can improve performance and ensure the database remains consistent with the real-world entities it represents.

2. CREATE a table called Artists.

CREATE TABLE Artists (

artist\_id NUMBER PRIMARY KEY,

first\_name VARCHAR2(50),

last\_name VARCHAR2(50),

band\_name VARCHAR2(100),

email VARCHAR2(100),

hourly\_rate NUMBER

);

a. Add the following to the table:

• artist ID

• first name

• last name

• band name

• email

• hourly rate

b. INSERT one artist from the d\_songs table.

INSERT INTO Artists (artist\_id, first\_name, last\_name, band\_name, email, hourly\_rate)

SELECT artist\_id, first\_name, last\_name, band\_name, email, hourly\_rate

FROM d\_songs

WHERE artist\_id = 1; -- Replace with actual artist\_id from d\_songs

c. INSERT one artist of your own choosing.

INSERT INTO Artists (artist\_id, first\_name, last\_name, band\_name, email, hourly\_rate)

VALUES (2, 'John', 'Doe', 'Rock Legends', 'johndoe@example.com', 150);

d. Give an example how each of the following may be used on the table that you have created:

1) ALTER TABLE

ALTER TABLE Artists ADD (genre VARCHAR2(50));

2) DROP TABLE

DROP TABLE Artists;

3) RENAME TABLE

RENAME Artists TO Music\_Artists;

4) TRUNCATE

TRUNCATE TABLE Artists;

5) COMMENT ON TABLE

COMMENT ON TABLE Artists IS 'Table contains data about music artists';

3. In your o\_employees table, enter a new column called “Termination.” The datatype for the new column should be VARCHAR2. Set the DEFAULT for this column as SYSDATE to appear as character data in the format: February 20th, 2003.

4. Create a new column in the o\_employees table called start\_date. Use the TIMESTAMP WITH LOCAL TIME ZONE as the datatype.

5. Truncate the o\_jobs table. Then do a SELECT \* statement. Are the columns still there? Is the data still there?

6. What is the distinction between TRUNCATE, DELETE, and DROP for tables?

TRUNCATE is used for fast data removal with no need for logging each row, typically for clearing a table.

DELETE is used when you need to selectively remove rows based on conditions or when logging and triggers are important.

DROP is used when you want to permanently remove both the table and its data from the database entirely.

7. List the changes that can and cannot be made to a column.

CAN change: Modify column name, modify data type, increase or decrease size, change null/not null constraint, add/drop a default value

CANNOT be changed: change column type with incompatible data types, change column data type length below existing data, change column type when data requires conversion, change a column to or from an auto-increment (identity) column, change the column order, change the column type when it breaks existing constraints

8. Add the following comment to the o\_jobs table:

"New job description added"

View the data dictionary to view your comments.

9. Rename the o\_jobs table to o\_job\_description.

10. F\_staffs table exercises:

a. Create a copy of the f\_staffs table called copy\_f\_staffs and use this copy table for the remaining labs in this lesson.

b. Describe the new table to make sure it exists.

c. Drop the table.

d. Try to select from the table.

e. Investigate your recyclebin to see where the table went.

f. Try to select from the dropped table by using the value stored in the OBJECT\_NAME column. You will need to copy and paste the name as it is exactly, and enclose the new name in “ “ (double quotes). So if the dropped name returned to you is BIN$Q+x1nJdcUnngQESYELVIdQ==$0, you need to write a query that refers to “BIN$Q+x1nJdcUnngQESYELVIdQ==$0”.

g. Undrop the table.

h. Describe the table.

11. Still working with the copy\_f\_staffs table, perform an update on the table.

a. Issue a select statement to see all rows and all columns from the copy\_f\_staffs table;

b. Change the salary for Sue Doe to 12 and commit the change.

c. Issue a select statement to see all rows and all columns from the copy\_f\_staffs table;

d. For Sue Doe, update the salary to 2 and commit the change.

e. Issue a select statement to see all rows and all columns from the copy\_f\_staffs table;

f. Now, issue a FLASHBACK QUERY statement against the copy\_f\_staffs table, so you can see all the changes made.

g. Investigate the result of f), and find the original salary and update the copy\_f\_staffs table salary column for Sue Doe back to her original salary.