

Practice 12-1

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 into	Adds a new row to a table

1. Give two examples of why it is important to be able to alter the data in database.
 - a. Update patient records in EMR
 - b. Adjust inventory level
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	Produce	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

3. DJs on Demand has two new events coming up. One event is a fall football part 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

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

Client_Number	First_Name	Last_Name	Phone	email
6655	Ayako	Dahish	3608859030	dahisha@harbot.net
6689	Nick	Neuville	9048953049	nnick@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.

ID	Name	Event_Date	Description	Cost	Venue_ID	Pakcage_Code	Theme_Code	Client_Number
110	Ayako Anniversary	07-Jul-2004	Part 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

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.

update	Modifies existing rows in a table
Correlated subquery update	Retrieves information from one table & uses the information to update another table
Integrity constraints	Ensures that the data adheres to a predefined set of rules
Correlated subquery delete	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.

```
CREATE TABLE copy_f_food_items
AS ( SELECT * FROM f_food_items);
DESCRIBE f_food_items;
DESCRIBE copy_f_food_items;
SELECT * FROM f_food_items;
SELECT * FROM copy_f_food_items;

UPDATE copy_f_food_items SET price = 3.75
WHERE LOWER(description) = 'strawberry shake';
UPDATE copy_f_food_items SET price = 1.20
WHERE LOWER(description) = 'fries';

SELECT * FROM copy_f_food_items;
```

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?)

```
CREATE TABLE copy_f_staffs
AS ( SELECT * FROM f_staffs);
DESCRIBE f_staffs;
DESCRIBE copy_f_staffs;
SELECT * FROM f_staffs;
SELECT * FROM copy_f_staffs;
```

```
UPDATE copy_f_staffs SET overtime_rate = NVL(overtime_rate, 0) + 0.75
WHERE LOWER(first_name || ' ' || last_name) = 'bob miller';
UPDATE copy_f_staffs SET overtime_rate = NVL(overtime_rate, 0) + 0.85
WHERE LOWER(first_name || ' ' || last_name) = 'sue doe';
```

```
SELECT * FROM copy_f_staffs;
```

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

ORDER_NUMB ER	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

```
CREATE TABLE copy_f_orders
AS ( SELECT * FROM f_orders);
DESCRIBE f_orders;
DESCRIBE copy_f_orders;
SELECT * FROM f_orders;
SELECT * FROM copy_f_orders;
```

```
INSERT INTO copy_f_orders(order_number,order_date,order_total,cust_id,staff_id)
VALUES(5680,TO_DATE('June 12, 2004','fmMonth dd, yyyy'),159.78,145,9);
INSERT INTO copy_f_orders(order_number,order_date,order_total,cust_id,staff_id)
VALUES(5691,TO_DATE('09-23-2004','mm-dd-yyyy'),145.98,225,12);
INSERT INTO copy_f_orders(order_number,order_date,order_total,cust_id,staff_id)
VALUES(5701,TO_DATE('July 4, 2004','fmMonth dd, yyyy'),229.31,230,12);
```

```
SELECT * FROM copy_f_orders;
```

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 all these records successfully? Yes

ID	FIRST_NAM E	LAST_NAM E	ADDRESS	CITY	STATE	ZIP	PHONE_NU MBER
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
LOWER(first_name || ' ' || last_name) = 'bob miller')
WHERE LOWER(first_name || ' ' || last_name) = 'sue doe';
SELECT * from copy_f_staffs;
```

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

INSERT INTO

```
copy_f_staffs(id,first_name,last_name,birthdate,salary,overtime_rate,training,staff_type,manager_id,manager_budget,manager_target)
VALUES(25,'Kai','Kim',TO_DATE('03-Nov-1988','fmdd-Mon-yyyy'),6.75,NULL,NULL,'Order
Taker',NULL,NULL,NULL);
```

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 LOWER(first_name || ' ' || last_name) = 'sue doe')
WHERE LOWER(first_name || ' ' || last_name) = 'kai kim';
```

```
SELECT * FROM copy_f_staffs;
```

8. Execute the following SQL statement. Record your results.

```
DELETE from departments
```

WHERE department_id = 60;

ORA-02292: integrity constraint (HKUMAR.EMP_DEPT_FK) violated - child record found

9. Kim Kaihas 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.

```
SELECT * FROM copy_f_staffs;
DELETE FROM copy_f_staffs
WHERE LOWER(first_name || ' ' || last_name) = 'kai kim';
SELECT * FROM copy_f_staffs;
```

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);
```

```
DESCRIBE employees;
DESCRIBE lesson7_emp;
SELECT * FROM employees;
SELECT * FROM lesson7_emp;
```

```
SELECT DISTINCT employee_id FROM job_history;
7 rows returned in 0.00 seconds
```

```
DELETE FROM lesson7_emp
WHERE employee_id IN ( SELECT DISTINCT employee_id FROM job_history) ;
```

5 row(s) deleted.

Practice 12-3

1. When would you want a DEFAULT value?
Streamline data and establish baseline for logic models such as a “pending” status
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?
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.
- Assign new cd_numbers to each new CD acquired.
 - 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);  
DESCRIBE d_cds;  
DESCRIBE manager_copy_d_cds;  
SELECT * FROM d_cds;  
SELECT * FROM manager_copy_d_cds;
```

- c. INSERT into the manger_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', and '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');
```

```
INSERT INTO manager_copy_d_cds(cd_number,title,producer,year)  
VALUES(97,'Celebrate the Day','R & B Inc.','2003');
```

```
INSERT INTO manager_copy_d_cds(cd_number,title,producer,year)  
VALUES(99,'Party Music','Old Town Records','2004');
```

```
INSERT INTO manager_copy_d_cds(cd_number,title,producer,year)  
VALUES(100,'Best of Rock and Roll','Old Town Records','2004');
```

- 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?

```
DELETE FROM manager_copy_d_cds  
WHERE cd_number = 91;
```

```
SELECT * FROM manager_copy_d_cds ;
```

```
MERGE INTO manager_copy_d_cds tgt USING d_cds src  
ON (src.cd_number = tgt.cd_number)  
WHEN MATCHED THEN UPDATE  
SET tgt.title = src.title, tgt.producer = src.producer, tgt.year = src.year  
WHEN NOT MATCHED THEN INSERT  
VALUES (src.cd_number, src.title, src.producer, src.year)
```

4. Run the follow 3 statements to create 2 nwq tables for us in a Multi-table insert statement. All 3 tables should be empty on create, 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

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	Specifies a preset value of a value is omitted in the INSERT statement
table	Stores data; basic unit of storage composed of rows and columns
Create table	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.
2. Write the syntax to create the grad_candidates table.

```
CREATE TABLE graduate_candidates
( student_id NUMBER(6,0),
  last_name VARCHAR2(75) CONSTRAINT gcs_last_name_nn NOT NULL ENABLE,
  first_name VARCHAR2(75) CONSTRAINT gcs_first_name_nn NOT NULL ENABLE,
  credits NUMBER(5,2) CONSTRAINT gcs_credits_nn NOT NULL ENABLE,
  graduation_date DATE,
  CONSTRAINT gcs_std_id_pk PRIMARY KEY (student_id),
  CONSTRAINT gcs_req_fk FOREIGN KEY(credits) REFERENCES requirements(credits) ENABLE );
```

3. Confirm creation of the table using DESCRIBE.
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.
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_CATALOGUE or USER_CAT

In separate sentences, summarize what each query will return.

Practice 13-2

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

Timestamp with local time zone	When a column is selected in a SQL statement the time is automatically converted to the user's timezone
blob	Binary large object data up to 4 gigabyte
Timestamp with time zone	Stores a time zone value as a displacement from Universal Coordinated Time or UCT
clob	Allows time to be stored as an interval of days to hours, minute, and seconds
cclob	Character data up to 4 gigabyte
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
 - b. INTERVAL YEAR TO MONTH
 - c. INTERVAL DAY TO SECOND
2. Execute a SELECT* from each table to verify your input
3. Give 3 examples of organizations and personal situations where it is important to know to which time zone a date-time value refers.
Organize international meeting, personal travel (international), e-commerce sites

Practice 13-3

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.

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

- a. Add new features, updating data, and optimizing db performance
6. CREATE a table called Artists.
 - a. Add the following to the table:
 - artist ID
 - first name
 - last name
 - band name
 - email
 - hourly rate

```
CREATE TABLE artists (artist_id NUMBER(5,0),
first_name VARCHAR2(25) CONSTRAINT ait_first_name_nn NOT NULL ENABLE,
last_name VARCHAR2(30) CONSTRAINT ait_last_name_nn NOT NULL ENABLE,
band_name VARCHAR2(30),
email VARCHAR2(75) CONSTRAINT ait_email_nn NOT NULL ENABLE,
hr_rate NUMBER(8,2) CONSTRAINT ait_hr_rate_nn NOT NULL ENABLE,
song_id NUMBER(5,0) CONSTRAINT ait_song_id_nn NOT NULL ENABLE,
CONSTRAINT ait_id_pk PRIMARY KEY (artist_id)
)
```

B. INSERT one artist from the d_songs table.

```
INSERT INTO artists (artist_id, first_name, last_name, band_name, email, hr_rate, song_id)
SELECT 1 AS artist_id,
CASE
WHEN artist IS NULL THEN 'first name unknown'
WHEN INSTR(artist,' ') = 0 THEN artist
ELSE SUBSTR(artist,1,INSTR(artist,' ') -1)
END
AS first_name,
CASE
WHEN artist IS NULL THEN 'last name unknown'
WHEN INSTR(artist,' ') = 0 THEN artist
ELSE SUBSTR(artist,INSTR(artist,' '),LENGTH(artist))
END
AS last_name,
artist as band_name,
NULL as email,
NULL as hr_rate,
id as song_id

FROM d_songs
WHERE ROWNUM =1 ;

SELECT * FROM artists;
```

C. INSERT one artist of your own choosing.

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

- 1) ALTER TABLE
- 2) DROP TABLE
- 3) RENAME TABLE
- 4) TRUNCATE
- 5) COMMENT ON TABLE

7. 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.
8. Create a new column in the o_employees table called start_date. Use the TIMESTAMP WITH LOCAL TIME ZONE as the datatype.
9. Truncate the o_jobs table. Then do a SELECT * statement.
Are the columns still there? Is the data still there?

```
DESCRIBE o_jobs;
SELECT * FROM o_jobs;
TRUNCATE TABLE o_jobs;
DESCRIBE o_jobs;
SELECT * FROM o_jobs;
Yes, but the data is gone.
```

10. What is the distinction between TRUNCATE, DELETE, and DROP for tables?
 - a. Truncate is much faster than delete for large table, it deletes all rows in a table, but keeps the table and its structure
 - b. Delete: deletes specific rows or all depends on condition, its used to delete some data
 - c. Drop: completely removes table from db, it is irreversible and cannot selectively remove data.
11. List the changes that can and cannot be made to a column.
 - a. Can: change data type, add or remove default value, rename column
 - b. Cannot: change primary key attribute, reduce column size without data loss, change column's type
12. Add the following comment to the o_jobs table:
"New job description added"
View the data dictionary to view your comments.
13. Rename the o_jobs table to o_job_description.
14. 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.

```
CREATE TABLE copy_f_staffs
AS ( SELECT * FROM f_staffs);
DESCRIBE f_staffs;
DESCRIBE copy_f_staffs;
SELECT * FROM f_staffs;
SELECT * FROM copy_f_staffs;
```

- B. Describe the new table to make sure it exists.

```
DESC copy_f_staffs;
```

- C. Drop the table.

```
DROP TABLE copy_f_staffs;
```

- D. Try to select from the table.

```
SELECT * FROM copy_f_staffs;
```

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

```
DESCRIBE user_recyclebin ;
SELECT * FROM
  (SELECT * FROM user_recyclebin ORDER BY droptime DESC)
  WHERE ROWNUM <= 100;
SELECT object_name,droptime FROM user_recyclebin WHERE LOWER(original_name) =
'copy_f_staffs';
```

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+x1nJdcUnngQESYELVldQ==\$0, you need to write a query that refers to "BIN\$Q+x1nJdcUnngQESYELVldQ==\$0".

```
SELECT * FROM "BIN$QF30ctmEV7jgU81jFJDpGA==$0";
```

G. Undrop the table.

```
FLASHBACK TABLE copy_f_staffs TO BEFORE DROP;
```

H. Describe the table.

```
DESCRIBE copy_f_staffs;
```

15. 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;

```
SELECT * FROM copy_f_staffs;
```

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

```
UPDATE copy_f_staffs
SET salary = 12
WHERE first_name = 'Sue' AND last_name = 'Doe';
```

C. Issue a select statement to see all rows and all columns from the copy_f_staffs table;

```
SELECT * FROM copy_f_staffs;
```

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

```
UPDATE copy_f_staffs
SET salary = 2
WHERE first_name = 'Sue' AND last_name = 'Doe';
```

E. Issue a select statement to see all rows and all columns from the copy_f_staffs table;

```
SELECT * FROM copy_f_staffs;
```

F. Now, issue a FLASHBACK QUERY statement against the copy_f_staffs table, so you can see all the changes made.

```
SELECT versions_operation, versions_starttime, versions_endtime, id, first_name, last_name, birthdate,
salary,overtime_rate,training,staff_type,manager_id, manager_budget,manager_target
FROM copy_f_staffs
VERSIONS BETWEEN SCN MINVALUE AND MAXVALUE
WHERE id = 12;
```

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.

```
UPDATE copy_f_staffs
SET salary = (SELECT salary
FROM copy_f_staffs
WHERE first_name = 'Sue' AND last_name = 'Doe' AND versions_operation IS NULL AND
versions_starttime IS NULL)
WHERE first_name = 'Sue' AND last_name = 'Doe';
```

```
SELECT * FROM copy_f_staffs;
SELECT versions_operation, versions_starttime, versions_endtime, id, first_name, last_name, birthdate,
salary,overtime_rate,training,staff_type,manager_id, manager_budget,manager_target
FROM copy_f_staffs
```