**Programming 14+**

|  |  |
| --- | --- |
| Instructions |  |
| • 14-1 Intro to Constraints; NOT NULL and UNIQUE Constraints  • 14-2 PRIMARY KEY, FOREIGN KEY, and CHECK Constraints  • 14-3 Managing Constraints  • 15-1 Creating Views  • 15-2 DML Operations and Views  • 15-3 Managing Views  • 16-1 Working With Sequences  • 16-2 Indexes and Synonyms  • 17-1 Controlling User Access  • 17-2 Creating and Revoking Object Privileges  • 17-3 Regular Expressions | |

Database Programming with SQL  
14-1: Intro to Constraints; NOT NULL and UNIQUE Constraints  
Practice Activities  
Objectives  
• Define the term "constraint" as it relates to data integrity  
• State when it is possible to define a constraint at the column level, and when it is possible at the table level  
• State why it is important to give meaningful names to constraints  
• State which data integrity rules are enforced by NOT NULL and UNIQUE constraints  
• Write a CREATE TABLE statement which includes NOT NULL and UNIQUE constraints at the table and column levels  
• Explain how constraints are created at the time of table creation

Vocabulary  
Identify the vocabulary word for each definition below.

* Every value in a column or set of columns (a composite key)  
  must be unique  
  --unique constraint
* For every row entered into the table, there must be a value for  
  that column  
  --not null constraint
* Constraint ensures that the column contains no null values and  
  uniquely identifies each row of the table  
  --primary key
* Specifies a condition for a column that must be true for each row  
  of data  
  --check constraint
* Identifies that table and column in the parent table  
  --foreign key
* An integrity constraint that requires every value in a column or set  
  of columns be unique  
  -- unique constraint
* Designates a column (child table) that establishes a relationship  
  between a primary key in the same table and a different table  
  (parent table)  
  --foreign key
* References one or more columns and is defined separately from  
  the definitions of the columns in the table  
  -- table level constraint
* Database rule.  
  -- constraint
* Database rule that references a single column

-- column level constraint  
Try It / Solve It  
Global Fast Foods has been very successful this past year and has opened several new stores. They need to add a table to their database to store information about each of their store’s locations. The owners want to make sure that all entries have an identification number, date opened, address, and city and that no other entry in the table can have the same email address. Based on this information, answer the following questions about the global\_locations table. Use the table for your answers.

1. What is a “constraint” as it relates to data integrity?

-- A constraint is a restriction that is utilized to ensure data remains relevant and correct.

2. What are the limitations of constraints that may be applied at the column level and at the table level?

-- There are limitations of constraints at both the column level and table level. Column level constraint includes single focus on one column only ; Table level constraint includes not being able to focus on single columns. Table level can also be more complex due to longer SQL length.

3. Why is it important to give meaningful names to constraints?  
-- To be able to comprehend easily and allow for easy ability to make adjustments/updates as needed

4. Based on the information provided by the owners, choose a datatype for each column. Indicate the length, precision, and scale for each NUMBER datatype.

5. Use “nullable” to indicate those columns that can have null values.

CREATE TABLE global\_locations (

id NUMBER(5) CONSTRAINT global\_locations\_id\_pk PRIMARY KEY NOT NULL,

name VARCHAR2(100) NOT NULL,

date\_opened DATE NOT NULL,

address VARCHAR2(200) NOT NULL,

city VARCHAR2(50) NOT NULL,

zip\_postal\_code VARCHAR2(15) NULL,

phone VARCHAR2(15) NULL,

email VARCHAR2(100) CONSTRAINT global\_locations\_email\_uk UNIQUE NULL,

manager\_id NUMBER(5) NULL,

emergency\_contact VARCHAR2(100) NULL

);

6. Write the CREATE TABLE statement for the Global Fast Foods locations table to define the constraints at the column level.

CREATE TABLE global\_locations (

id NUMBER(5) PRIMARY KEY,

name VARCHAR2(100) NOT NULL,

date\_opened DATE NOT NULL,

address VARCHAR2(200) NOT NULL,

city VARCHAR2(50) NOT NULL,

zip\_postal\_code VARCHAR2(15),

phone VARCHAR2(15),

email VARCHAR2(100) UNIQUE,

manager\_id NUMBER(5),

emergency\_contact VARCHAR2(100)

);

7. Execute the CREATE TABLE statement in Oracle Application Express.  
8. Execute a DESCRIBE command to view the Table Summary information.

DESCRIBE global\_locations;  
9. Rewrite the CREATE TABLE statement for the Global Fast Foods locations table to define the UNIQUE constraints at the table level. Do not execute this statement

CREATE TABLE global\_locations (

id NUMBER(5) PRIMARY KEY,

name VARCHAR2(100) NOT NULL,

date\_opened DATE NOT NULL,

address VARCHAR2(200) NOT NULL,

city VARCHAR2(50) NOT NULL,

zip\_postal\_code VARCHAR2(15),

phone VARCHAR2(15),

email VARCHAR2(100),

manager\_id NUMBER(5),

emergency\_contact VARCHAR2(100),

CONSTRAINT unique\_email UNIQUE (email)

);

Database Programming with SQL  
14-2: PRIMARY KEY, FOREIGN KEY, and CHECK Constraints  
Practice Activities  
Objectives  
• Define and give an example of PRIMARY KEY, FOREIGN KEY, and CHECK constraints  
• Explain the purpose of defining PRIMARY KEY, FOREIGN KEY, and CHECK constraints on a  
table  
• Demonstrate the creation of constraints at the column level and table level in a CREATE  
TABLE statement  
• Evaluate a business problem requiring the addition of a PRIMARY KEY and FOREIGN KEY  
constraint and write the code to execute the change

Vocabulary  
Identify the vocabulary word for each definition below.

* Allows a foreign key row that is referenced to a primary key row to be deleted  
  on delete cascade
* Explicitly defines a condition that must be met  
  check constraint
* A column or set of columns that uniquely identifies each row in a table  
  primary key
* Constraint ensures that the column contains no null values  
  not null constraint
* Allows a child row to remain in a table with null values when a parent record has been deleted  
  on delete set
* Establishes a relationship between the foreign key column and a primary key or unique key in the same table or a different table

Foreign key

Try It / Solve It  
1. What is the purpose of a  
a. PRIMARY KEY

-- unique identifier that ensures each row is identifiable.   
b. FOREIGN KEY

-- establish connection between two tables  
c. CHECK CONSTRAINT

-- ensure data is valid

2. Using the column information for the animals table below, name constraints where applicable at the table level, otherwise name them at the column level. Define the primary key (animal\_id). The license\_tag\_number must be unique. The admit\_date and vaccination\_date columns cannot  
contain null values.

animal\_id NUMBER(6)   
name VARCHAR2(25)  
license\_tag\_number NUMBER(10)  
admit\_date DATE  
adoption\_id NUMBER(5),  
vaccination\_date DATE

animal\_id NUMBER(6) = primary key

license\_tag\_number NUMBER(10) = unique id

admit\_date DATE = not null

vaccination\_date DATE = not null

3. Create the animals table. Write the syntax you will use to create the table.

CREATE TABLE animals ( animal\_id NUMBER(6) PRIMARY KEY, name VARCHAR2(25), license\_tag\_number NUMBER(10) UNIQUE, admit\_date DATE NOT NULL, adoption\_id NUMBER(5), vaccination\_date DATE NOT NULL

);

4. Enter one row into the table. Execute a SELECT \* statement to verify your input. Refer to the graphic below for input.

5. Write the syntax to create a foreign key (adoption\_id) in the animals table that has a  
corresponding primary- key reference in the adoptions table. Show both the column-level and table-level syntax. Note that because you have not actually created an adoptions table, no adoption\_id primary key exists, so the foreign key cannot be added to the animals table.

CREATE TABLE animals ( animal\_id NUMBER(6) PRIMARY KEY, name VARCHAR2(25), license\_tag\_number NUMBER(10) UNIQUE, admit\_date DATE NOT NULL, adoption\_id NUMBER(5) REFERENCES adoptions(adoption\_id), vaccination\_date DATE NOT NULL

);

6. What is the effect of setting the foreign key in the ANIMAL table as:  
a. ON DELETE CASCADE

--allows for all deletion   
b. ON DELETE SET NULL

-- allows for child table to be null when a parent table is deleted

7. What are the restrictions on defining a CHECK constraint?

-- only able to refer to columns within that table ; subqueries not allowed ; table level

Database Programming with SQL  
14-3: Managing Constraints  
Practice Activities  
Objectives  
• List four different functions that the ALTER statement can perform on constraints  
• Write ALTER TABLE statements to add, drop, disable, and enable constraints  
• Name a business function that would require a DBA to drop, enable, and/or disable a  
constraint or use the CASCADE syntax  
• Query the data dictionary for USER\_CONSTRAINTS and interpret the information returned

Vocabulary  
Identify the vocabulary word for each definition below.

- To deactivate an integrity constraint

-- disable constraint  
- Disables dependent integrity constraints

-- cascade   
- To add, modify, or drop columns from a table

-- alter table  
- To activate an integrity constraint currently disabled

-- enable constraint  
- Removes a constraint from a table

-- drop constraint  
- Allows user to delete a column from a table

-- drop column  
- Defines the actions the database server takes when a user  
attempts to delete or update a key to which existing foreign keys  
point

-- cascade constraint

Try It / Solve It  
Using Oracle Application Express, click the SQL Workshop tab in the menu bar. Click the Object Browser and verify that you have a table named copy\_d\_clients and a table named copy\_d\_events. If you don’t have these tables in your schema, create them before completing the exercises below. Here is how the original tables are related. The d\_clients table has a primary key client\_number. This has a primary-key constraint and it is referenced in the foreign-key constraint on the d\_events table.  
1. What are four functions that an ALTER statement can perform on constraints?

-- add, drop, enable, disable

2. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy\_d\_clients table. Name the primary key copy\_d\_clients\_pk . What is the syntax you used to create the PRIMARY KEY constraint to the copy\_d\_clients.table?

ALTER TABLE copy\_d\_clients ADD CONSTRAINT copy\_d\_clients\_pk PRIMARY KEY (client\_number);

3. Create a FOREIGN KEY constraint in the copy\_d\_events table. Name the foreign key  
copy\_d\_events\_fk. This key references the copy\_d\_clients table client\_number column. What is the syntax you used to create the FOREIGN KEY constraint in the copy\_d\_events table?

ALTER TABLE  copy\_d\_events

ADD CONSTRAINT copy\_d\_eve\_client\_number\_fk FOREIGN KEY (client\_number) REFERENCES  copy\_d\_clients (client\_number) ENABLE;

4. Use a SELECT statement to verify the constraint names for each of the tables. Note that the tablenames must be capitalized.  
a. The constraint name for the primary key in the copy\_d\_clients table is PRIMARY .  
b. The constraint name for the foreign key in the copy\_d\_events table is FOREIGN .

5. Drop the PRIMARY KEY constraint on the copy\_d\_clients table. Explain your results.

ALTER TABLE copy\_d\_clients DROP CONSTRAINT copy\_d\_clients\_pk;

6. Add the following event to the copy\_d\_events table. Explain your results.

7. Create an ALTER TABLE query to disable the primary key in the copy\_d\_clients table. Then add the values from #6 to the copy\_d\_events table. Explain your results.

INSERT INTO copy\_d\_events (client\_number, event\_date, event\_details) VALUES (12345, TO\_DATE('2024-11-22', 'YYYY-MM-DD'), 'Event 2');

8. Repeat question 6: Insert the new values in the copy\_d\_events table. Explain your results.

ALTER TABLE copy\_d\_clients ENABLE CONSTRAINT copy\_d\_clients\_pk;

9. Enable the primary-key constraint in the copy\_d\_clients table. Explain your results.

ALTER TABLE copy\_d\_events ENABLE CONSTRAINT copy\_d\_events\_fk;

10. If you wanted to enable the foreign-key column and reestablish the referential integrity between these two tables, what must be done?

ALTER TABLE copy\_d\_events ENABLE CONSTRAINT copy\_d\_events\_fk;

11. Why might you want to disable and then re-enable a constraint?  
-- ensure up to date and address potential issues

12. Query the data dictionary for some of the constraints that you have created. How does the data dictionary identify each constraint type?

-- c = check constraint

-- p = primary key constraint

-- u = unique key constraint

Database Programming with SQL  
15-1: Creating Views  
Practice Activities  
Objectives  
• List three uses for views from the standpoint of a database administrator  
• Explain, from a business perspective, why it is important to be able to create and use logical  
subsets of data derived from one or more tables  
• Create a view with and without column aliases in the subquery using a single base table  
• Create a complex view that contains group functions to display values from two tables  
• Retrieve data from a view

Vocabulary  
Identify the vocabulary word for each definition below.

* A subset of data from one or more tables that is generated from a query and stored as a virtual table  
  --view
* Name of view  
  -- view name
* Creates a view regardless of whether or not the base tables exist  
  -- create view
* Derives data from a table, no functions or groups, performs DML operations through the view  
  -- simple view
* Creates the view only if the base table exists  
  -- create view
* Statement used to create a new view  
  --create view
* Specifies a name for each expression selected by the view’s query  
  --column alias
* A complete SELECT statement  
  --view query
* Derives data from more than one table, contains functions or groups of data, and does not always allow DML operations through the view  
  --complex view
* Re-creates the view if it already exists

--create view

Try It / Solve It  
1. What are three uses for a view from a DBA’s perspective?

-- restrictions; decrease complexity; customization/personalization

2. Create a simple view called view\_d\_songs that contains the ID, title, and artist from the DJs on Demand table for each “New Age” type code. In the subquery, use the alias “Song Title” for the title column.

SELECT d\_songs.id, d\_songs.title "Song Title", d\_songs.artist

from d\_songs INNER JOIN d\_types ON d\_songs.type\_code = d\_types.code

where d\_types.description = 'New Age'

3. SELECT \*  
FROM view\_d\_songs.  
What was returned?  
-- id, song title, artist

4. REPLACE view\_d\_songs. Add type\_code to the column list. Use aliases for all columns.

CREATE OR REPLACE VIEW view\_d\_songs AS

SELECT d\_songs.id, d\_songs.title "Song Title", d\_songs.artist, d\_songs.type\_code

from d\_songs INNER JOIN d\_types ON d\_songs.type\_code = d\_types.code

where d\_types.description = 'New Age'

;

5. Jason Tsang, the disk jockey for DJs on Demand, needs a list of the past events and those  
planned for the coming months so he can make arrangements for each event’s equipment setup. As the company manager, you do not want him to have access to the price that clients paid for their events. Create a view for Jason to use that displays the name of the event, the event date, and the theme description. Use aliases for each column name.

CREATE OR REPLACE VIEW jason\_event\_view AS SELECT event\_name AS "Event Name", event\_date AS "Event Date", theme\_description AS "Theme" FROM events;

6. It is company policy that only upper-level management be allowed access to individual employee salaries. The department managers, however, need to know the minimum, maximum, and average salaries, grouped by department. Use the Oracle database to prepare a view that  
displays the needed information for department managers.

CREATE OR REPLACE VIEW dept\_salary\_stats AS SELECT department\_id AS "Department ID", MIN(salary) AS "Minimum Salary", MAX(salary) AS "Maximum Salary", AVG(salary) AS "Average Salary" FROM employees GROUP BY department\_id;

Database Programming with SQL  
15-2: DML Operations and Views  
Practice Activities  
Objectives  
• Write and execute a query that performs DML operations on a simple view  
• Name the conditions that restrict modifying a view using DML operations  
• Write and execute a query using the WITH CHECK OPTION clause  
• Explain the use of WITH CHECK OPTION as it applies to integrity constraints and data validation  
• Apply the WITH READ ONLY option to a view to restrict DML operations

Vocabulary  
Identify the vocabulary word for each definition below.  
- A pseudocolumn which assigns a sequential value starting with 1  
to each of the rows returned from the subquery

= rownum  
- Specifies that INSERTS and UPDATES performed through the  
view can’t create rows which the view cannot select

= with check option  
- xEnsures that no DML operations can be performed on this view  
= with read only

Try It / Solve It  
Use the DESCRIBE statement to verify that you have tables named copy\_d\_songs, copy\_d\_events, copy\_d\_cds, and copy\_d\_clients in your schema. If you don't, write a query to create a copy of each.  
1. Query the data dictionary USER\_UPDATABLE\_COLUMNS to make sure the columns in the base tables will allow UPDATE, INSERT, or DELETE. Use a SELECT statement. All table names in the data dictionary are stored in uppercase.

SELECT TABLE\_NAME, COLUMN\_NAME, UPDATABLE FROM USER\_UPDATABLE\_COLUMNS WHERE UPDATABLE = 'YES';

2. Use the CREATE or REPLACE option to create a view of all the columns in the copy\_d\_songs table called view\_copy\_d\_songs.

CREATE OR REPLACE VIEW view\_copy\_d\_songs AS SELECT \* FROM copy\_d\_songs;

3. Use view\_copy\_d\_songs to INSERT the following data into the underlying copy\_d\_songs table. Execute a SELECT \* from copy\_d\_songs to verify your DML command. See the graphic.

INSERT INTO view\_copy\_d\_songs(id, title, duration, artist, type\_code)

VALUES(88, 'Mello Jello', '2 min', 'The What', 4);

4. Create a view based on the DJs on Demand COPY\_D\_CDS table. Name the view  
read\_copy\_d\_cds. Select all columns to be included in the view. Add a WHERE clause to restrict  
the year to 2000. Add the WITH READ ONLY option.

CREATE OR REPLACE VIEW read\_copy\_d\_cds  AS

SELECT \*

FROM copy\_d\_cds

WHERE year = '2000'

WITH READ ONLY ;

5. Using the read\_copy\_d\_cds view, execute a DELETE FROM read\_copy\_d\_cds WHERE  
cd\_number = 90;

DELETE

6. Use REPLACE to modify read\_copy\_d\_cds. Replace the READ ONLY option with WITH CHECK OPTION CONSTRAINT ck\_read\_copy\_d\_cds. Execute a SELECT \* statement to verify that the view exists.

CREATE OR REPLACE VIEW read\_copy\_d\_cds  AS

SELECT \*

FROM copy\_d\_cds

WHERE year = '2000'

WITH CHECK OPTION CONSTRAINT ck\_read\_copy\_d\_cds;

7. Use the read\_copy\_d\_cds view to delete any CD of year 2000 from the underlying copy\_d\_cds.

DELETE FROM read\_copy\_d\_cds

WHERE year = '2000';

8. Use the read\_copy\_d\_cds view to delete cd\_number 90 from the underlying copy\_d\_cds table.

DELETE FROM read\_copy\_d\_cds

WHERE cd\_number = 90;

9. Use the read\_copy\_d\_cds view to delete year 2001 records.

DELETE FROM read\_copy\_d\_cds WHERE year = 2001;

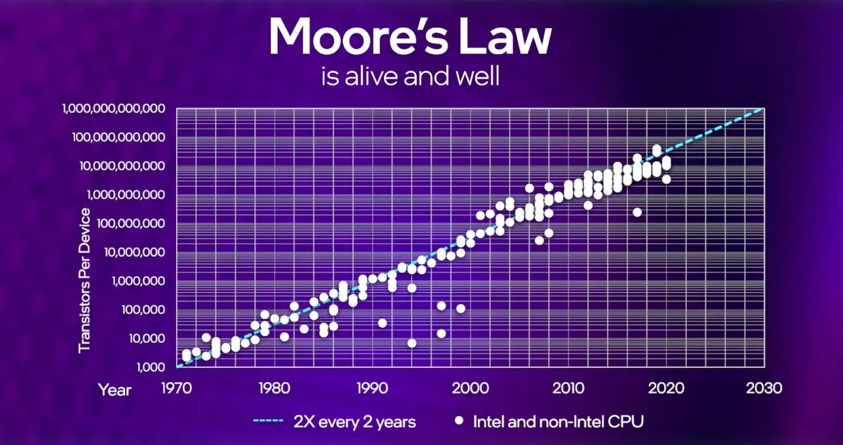
10. Execute a SELECT \* statement for the base table copy\_d\_cds. What rows were deleted?

The row that was deleted  
11. What are the restrictions on modifying data through a view?

--delete not possible on multi tables; cannot delete row if more than one base table is addressed

12. What is Moore’s Law? Do you consider that it will continue to apply indefinitely? Support your opinion with research from the internet.

-- Moore’s Law notes that the number of transistors on a microchip will double every 2 years, which notes that computing power will increase as time progresses.



13. What is the “singularity” in terms of computing?

-- potential future that notes that computer intelligence can advance past human intelligence

Database Programming with SQL  
15-3: Managing Views  
Practice Activities  
Objectives  
• Create and execute a query that removes a view  
• Create and execute a query using an inline view  
• Create and execute a top-n-analysis query

Vocabulary

Identify the vocabulary word for each definition below.

* Asks for the N largest or smallest values in a column  
  top n
* Removes a view  
  drop view
* Subquery with an alias that can be used within a SQL statement  
  inline view

Try It / Solve It  
1. Create a view from the copy\_d\_songs table called view\_copy\_d\_songs that includes only the title and artist. Execute a SELECT \* statement to verify that the view exists.  
2. Issue a DROP view\_copy\_d\_songs. Execute a SELECT \* statement to verify that the view has been deleted.  
3. Create a query that selects the last name and salary from the Oracle database. Rank the salaries from highest to lowest for the top three employees.  
4. Construct an inline view from the Oracle database that lists the last name, salary, department ID, and maximum salary for each department. Hint: One query will need to calculate maximum salary by department ID.  
5. Create a query that will return the staff members of Global Fast Foods ranked by salary from lowest to highest.

Extension Exercises  
1. Create a new table called my\_departments and add all columns and all rows to it using a  
subquery from the Oracle departments table. Do a SELECT \* from my\_departments to confirm that you have all the columns and rows.  
2. To view any constraints that may affect the my\_departments table, DESCRIBE my\_departments to check if any constraints were carried over from the departments table. If there are constraints on my\_departments, use an ALTER TABLE command to DISABLE all constraints on my\_departments.  
3. Create a view called view\_my\_departments that includes: department\_id and department\_name.  
4. Add the following data to the my\_departments table using view\_my\_departments.  
department\_id department\_name  
5. Create or enable the department\_id column as the primary key.  
6. Enter a new department named Human Resources into the my\_departments table using  
view\_my\_departments. Do not add a new department ID.  
7. Add the Human Resources department, department ID 220, to my\_departments using  
view\_my\_departments.  
8. Verify that the new additions to my\_departments were added using view\_my\_departments.  
See chart below  
9. Modify view\_my\_departments to include location ID. Do a SELECT \* command to show what columns are present and a DESCRIBE command to view the columns and associated constraints.  
10. Make location\_id a NOT NULL column in the my\_departments table.  
11. Using the Oracle database, create a complex view between locations and departments with only the following columns: department\_name, street\_address, city, and state. Include only U.S. cities.  
Verify that the view was created using a SELECT \* statement.  
See chart below

Database Programming with SQL  
16-1: Working with Sequences  
Practice Activities  
Objectives  
• List at least three useful characteristics of a sequence  
• Write and execute a SQL statement that creates a sequence  
• Query the data dictionary using USER\_SEQUENCES to confirm a sequence definition  
• Apply the rules for using NEXTVAL to generate sequential numbers for use in a table  
• List the advantages of caching sequence values  
• Name three reasons why gaps can occur in a sequence

Vocabulary  
Identify the vocabulary word for each definition below.

* Command that automatically generates sequential numbers
* Generates a numeric value
* Returns the next available sequence value
* Specifies the interval between sequence numbers
* Specifies a maximum value of 10^27 for an ascending sequence  
  and -1 for a descending sequence (default)
* returns the current sequence value
* specifies the minimum sequence value
* specifies whether the sequence continues to generate values after reaching its maximum or minimum values
* specifies a minimum value of 1 for an ascending sequence and –  
  (10^26) for a descending sequence (default)
* specifies a maximum or default value the sequence can generate
* specifies the first sequence number to be generated
* specifies how many values the Server pre-allocates and keeps in  
  memory

Try It / Solve It  
1. Using CREATE TABLE AS subquery syntax, create a seq\_d\_songs table of all the columns in the DJs on Demand database table d\_songs. Use the SELECT \* in the subquery to make sure that you have copied all of the columns.  
2. Because you are using copies of the original tables, the only constraints that were carried over were the NOT NULL constraints. Create a sequence to be used with the primary-key column of the seq\_d\_songs table. To avoid assigning primary-key numbers to these tables that already exist, the sequence should start at 100 and have a maximum value of 1000. Have your sequence increment by 2 and have NOCACHE and NOCYCLE. Name the sequence seq\_d\_songs\_seq.  
3. Query the USER\_SEQUENCES data dictionary to verify the seq\_d\_songs\_seq SEQUENCE  
settings.  
4. Insert two rows into the seq\_d\_songs table. Be sure to use the sequence that you created for the ID column. Add the two songs shown in the graphic.  
5. Write out the syntax for seq\_d\_songs\_seq to view the current value for the sequence. Use the DUAL table. (Oracle Application Developer will not run this query.)  
6. What are three benefits of using SEQUENCEs?  
7. What are the advantages of caching sequence values?  
8. Name three reasons why gaps may occur in a sequence?  
ID TITLE DURATION ARTIST TYPE\_CODE  
Island Fever 5 min Hawaiian Islanders 12  
Castle of Dreams 4 min The Wanderers 77

Extension Exercise  
1. Create a table called “students”. You can decide which columns belong in that table and what datatypes these columns require. (The students may create a table with different columns; however, the important piece that must be there is the student\_id column with a numeric datatype. This column length must allow the sequence to fit, e.g. a column length of 4 with a sequence that starts with 1 and goes to 10000000 will not work after student #9999 is entered.)  
2. Create a sequence called student\_id\_seq so that you can assign unique student\_id numbers for all students that you add to your table.  
3. Now write the code to add students to your STUDENTS table, using your sequence “database object.”

Database Programming with SQL  
16-2: Indexes and Synonyms  
Practice Activities  
Objectives  
• Define an index and its use as a schema object  
• Name the conditions that cause an index to be created automatically  
• Create and execute a CREATE INDEX and DROP INDEX statement  
• Construct and execute a function-based index  
• Construct a private and public synonym

Vocabulary  
Identify the vocabulary word for each definition below.

* Confirms the existence of indexes from the USER\_INDEXES data dictionary view
* Schema object that speeds up retrieval of rows
* To refer to a table by another name to simplify access
* An index that you create on multiple columns in a table
* The Oracle Server automatically creates this index when you  
  define a column in a table to have a PRIMARY KEY or a UNIQUE KEY constraint
* Stores the indexed values and uses the index based on a SELECT statement to retrieve the data
* Removes an index
* Gives alternative names to objects

Try It / Solve It  
1. What is an index and what is it used for?  
2. What is a ROWID, and how is it used?  
3. When will an index be created automatically?  
4. Create a nonunique index (foreign key) for the DJs on Demand column (cd\_number) in the D\_TRACK\_LISTINGS table. Use the Oracle Application Developer SQL Workshop Data Browser to confirm that the index was created.  
5. Use the join statement to display the indexes and uniqueness that exist in the data dictionary for the DJs on Demand D\_SONGS table.  
6. Use a SELECT statement to display the index\_name, table\_name, and uniqueness from the data dictionary USER\_INDEXES for the DJs on Demand D\_EVENTS table.  
7. Write a query to create a synonym called dj\_tracks for the DJs on Demand d\_track\_listings table.  
8. Create a function-based index for the last\_name column in DJs on Demand D\_PARTNERS table that makes it possible not to have to capitalize the table name for searches. Write a SELECT statement that would use this index.  
9. Create a synonym for the D\_TRACK\_LISTINGS table. Confirm that it has been created by  
querying the data dictionary.  
10. Drop the synonym that you created in question 9.

Database Programming with SQL  
17-1: Controlling User Access  
Practice Activities  
Objectives  
• Compare the difference between object privileges and system privileges  
• Construct the two commands required to enable a user to have access to a database  
• Construct and execute a GRANT... ON ...TO statement to assign privileges to objects in a  
user’s schema to other users and/or PUBLIC  
• Query the data dictionary to confirm privileges granted

Try It / Solve It  
1. What are system privileges concerned with?  
2. What are object privileges concerned with?  
3. What is another name for object security?  
4. What commands are necessary to allow Scott access to the database with a password of tiger?  
5. What are the commands to allow Scott to SELECT from and UPDATE the d\_clients table?  
6. What is the command to allow everybody the ability to view the d\_songs table?  
7. Query the data dictionary to view the object privileges granted to you the user.  
8. What privilege should a user be given to create tables?  
9. If you create a table, how can you pass along privileges to other users just to view your table?  
10. What syntax would you use to grant another user access to your copy\_employees table?  
11. How can you find out what privileges you have been granted for columns in the tables belonging to others?

Database Programming with SQL  
17-2: Creating and Revoking Object Privileges  
Practice Activities  
Objectives  
• Explain what a ROLE is and what its advantages are  
• Construct a statement to create a ROLE and GRANT privileges to it  
• Construct a GRANT ON TO WITH GRANT OPTION statement to assign privileges to objects in a user’s schema to other users and/or PUBLIC  
• Construct and execute a statement to REVOKE object privileges from other users and/or from PUBLIC  
• Distinguish between privileges and roles  
• Explain the purposes of a database link

Try It / Solve It  
1. What is a role?  
2. What are the advantages of a role to a DBA?  
3. Give the ability to another user in your class to look at one of your tables. Give him the right to let  
other students have that ability.  
4. You are the DBA. You are creating many users who require the same system privileges. What should you use to make your job easier?  
5. What is the syntax to accomplish the following?  
a. Create a role of manager that has the privileges to select, insert, and update and delete  
from the employees table  
b. Create a role of clerk that just has the privileges of select and insert on the employees  
table  
c. Grant the manager role to user scott  
d. Revoke the ability to delete from the employees table from the manager role  
6. What is the purpose of a database link?

Database Programming with SQL  
17-3: Regular Expressions  
Practice Activities  
Objectives  
• Describe regular expressions  
• Use regular expressions to search, match, and replace strings in SQL statements  
• Construct and execute regular expressions and check constraints

Try It / Solve It  
1. Working with the employees table, and using regular expressions, write a query that returns employees whose first names start with a “S” (uppercase) followed by either a “t” (lowercase) or “h” (lowercase).  
2. Investigate the LOCATIONS table.  
a. Describe the table.  
b. Perform a select that returns all rows and all columns of that table.  
c. Write a query using regular expressions that removes the spaces in the street\_address  
column in the LOCATIONS table.