

## Database Programming with SQL 14-1: Intro to Constraints; NOT NULL and UNIQUE Constraints Practice Activities

- **Vocabulary:** Identify the vocabulary word for each definition below
  - Every value in a column or set of columns (a composite key) must be unique
    - Answer: Unique constraint
  - For every row entered into the table, there must be a value for that column
    - Answer: Not NULL constraint
  - Constraint ensures that the column contains no null values and uniquely identifies each row of the table
    - Answer: Primary Key Constraint
  - Specifies a condition for a column that must be true for each row of data
    - Answer: Check Constraint
  - Identifies that table and column in the parent table
    - Answer: Foreign Key
  - An integrity constraint that requires every value in a column or set of columns be unique
    - Answer: 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)
    - Answer: Foreign Key Constraint
  - References one or more columns and is defined separately from the definitions of the columns in the table
    - Answer: Out-of-line Constraint
  - Database rule
    - Answer: Constraint
  - Database rule that references a single column
    - Answer: Column-level Constraint
- **Try It / Solve It**

Global Fast Foods global_locations Table						
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
Id						
name						
date_opened						
address						
city						
zip/postal code						
phone						
email						
manager_id						
Emergency contact						

- 1. What is a “constraint” as it relates to data integrity?
  - In the context of data integrity, constraints are rules or conditions that ensure and maintain the accuracy and reliability of data stored in a database
- 2. What are the limitations of constraints that may be applied at the column level and at the table level?
  - Column-level constraints apply to individual columns and are limited in their scope, while table-level constraints apply complex rules and relationships across multiple columns
- 3. Why is it important to give meaningful names to constraints?
  - It is important to give meaningful constraints names for understanding, maintaining and managing data integrity in databases
- 4. Based on the information provided by the owners, choose a data type for each column. Indicate the length, precision, and scale for each NUMBER datatype.
  - VARCHAR2: Variable-length text, length specified in characters
  - CHAR: Fixed-length text, usually for codes or short values
  - NUMBER: Numeric data, specified precision and scale
  - DATE: Date and time with fixed-length
- 5. Use “nullable” to indicate those columns that can have null values
  - ```
CREATE TABLE employees (
    employee_id NUMBER(6) CONSTRAINT pk_employee_id
    PRIMARY KEY,
    first_name VARCHAR2(50) NOT NULL,
    last_name VARCHAR2(50),
    hire_date DATE,
    salary NUMBER(10, 2) DEFAULT 0 NOT NULL
);
```
- 6. Write the CREATE TABLE statement for the Global Fast Foods locations table to define the constraints at the column level
  - ```
CREATE TABLE locations (
    location_id NUMBER(6) CONSTRAINT pk_location_id
    PRIMARY KEY,
    location_name VARCHAR2(100) NOT NULL,
    address VARCHAR2(255),
    city VARCHAR2(50) NOT NULL,
    postal_code VARCHAR2(20),
    country VARCHAR2(50) NOT NULL,
    latitude NUMBER(9, 6),
    longitude NUMBER(9, 6),
    created_date DATE DEFAULT SYSDATE NOT NULL
```

- ```
);
```
- 7. Execute the CREATE TABLE statement in Oracle Application Express
    - CREATE TABLE locations (
 

```
location_id NUMBER(6) CONSTRAINT pk_location_id
PRIMARY KEY,
location_name VARCHAR2(100) NOT NULL,
address VARCHAR2(255),
city VARCHAR2(50) NOT NULL,
postal_code VARCHAR2(20),
country VARCHAR2(50) NOT NULL,
latitude NUMBER(9, 6),
longitude NUMBER(9, 6),
created_date DATE DEFAULT SYSDATE NOT NULL
);
```
  - 8. Execute a DESCRIBE command to view the Table Summary information
    - DESCRIBE table\_name;
  - 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 locations (
 

```
location_id NUMBER(6) CONSTRAINT pk_location_id
PRIMARY KEY,
location_name VARCHAR2(100) NOT NULL,
address VARCHAR2(255),
city VARCHAR2(50) NOT NULL,
postal_code VARCHAR2(20),
country VARCHAR2(50) NOT NULL,
latitude NUMBER(9, 6),
longitude NUMBER(9, 6),
created_date DATE DEFAULT SYSDATE NOT NULL,
CONSTRAINT uq_location_name UNIQUE (location_name),
CONSTRAINT uq_postal_code UNIQUE (postal_code)
);
```

## Database Programming with SQL 14-2: PRIMARY KEY, FOREIGN KEY, and CHECK Constraints Practice Activities

- 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
    - Answer: ON DELETE CASCADE
  - Explicitly defines a condition that must be met

- Answer: Check Constraint
- A column or set of columns that uniquely identifies each row in a table
  - Answer: Primary Key
- Constraint ensures that the column contains no null values
  - Answer: Not NULL constraint
- Allows a child row to remain in a table with null values when a parent record has been deleted
  - Answer: ON DELETE SET NULL
- Establishes a relationship between the foreign key column and a primary key or unique key in the same table or a different table
  - Answer: Foreign Key Constraint
- Try It / Solve It
  - 1. What is the purpose of a
    - a. PRIMARY KEY
      - It is a unique identifier for a table
    - b. FOREIGN KEY
      - Establishes a relationship between two tables
    - c. CHECK CONSTRAINT
      - Ensure values meet specific conditions
  - 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.
    - **Table-Level Constraints:**
      - PRIMARY KEY (animal\_id)
      - UNIQUE (license\_tag\_number)
    - **Column-Level Constraints:**
      - 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

| ANIMAL_<br>ID | NAME | LICENSE_TAG_<br>NUMBER | ADMIT_DATE  | ADOPTION_<br>ID | VACCINATION_<br>DATE |
|---------------|------|------------------------|-------------|-----------------|----------------------|
| 101           | Spot | 35540                  | 10-Oct-2004 | 205             | 12-Oct-2004          |

- INSERT INTO animals (animal\_id, name, license\_tag\_number, admit\_date, adoption\_id, vaccination\_date)  
VALUES (101, 'Buddy', 1234567890, TO\_DATE('2024-11-01', 'YYYY-MM-DD'), 201, TO\_DATE('2024-11-15', 'YYYY-MM-DD'));
- 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
    - Records deleted in the parent table will be automatically deleted for any related records in the child table
  - b. ON DELETE SET NULL
    - Records deleted in the parent table will cause the foreign keys in the child table to become null but the records in the child table will remain
- 7. What are the restrictions on defining a CHECK constraint?
  - For CHECK constraints, each row in the table must satisfy the condition with either true or unknown

### Database Programming with SQL 14-3: Managing Constraints Practice Activities

- Vocabulary: Identify the vocabulary word for each definition below
  - To deactivate an integrity constraint
    - Answer: Disable Constraint
  - Disables dependent integrity constraints
    - Answer: CASCADE constraint disable
  - To add, modify, or drop columns from a table
    - Answer: Alter Table
  - To activate an integrity constraint currently disabled

- Answer: Enable constraint
- Removes a constraint from a table
  - Answer: Drop Constraint
- Allows user to delete a column from a table
  - Answer: 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
  - Answer: On delete/on update clause
- Try It / Solve It
  - 1. What are four functions that an ALTER statement can perform on constraints?
    - 1. Constraint
    - 2. Drop
    - 3. Modify
    - 4. Rename
  - 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\_events\_fk FOREIGN KEY (client\_number)  
REFERENCES copy\_d\_clients (client\_number);
  - 4. Use a SELECT statement to verify the constraint names for each of the tables. Note that the table names must be capitalized
    - a. The constraint name for the primary key in the copy\_d\_clients table is **primary key**
    - b. The constraint name for the foreign key in the copy\_d\_events table is **foreign key**
  - 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;

- Explanation: The client number column no longer have an enforced primary key
- 6. Add the following event to the copy\_d\_events table. Explain your results

| ID  | NAME              | EVENT_DATE  | DESCRIPTION                    | COST | VENUE_ID | PACKAGE_CODE | THEME_CODE | CLIENT_NUMBER |
|-----|-------------------|-------------|--------------------------------|------|----------|--------------|------------|---------------|
| 140 | Cline Bas Mitzvah | 15-Jul-2004 | Church and Private Home formal | 4500 | 105      | 87           | 77         | 7125          |

- INSERT INTO  
copy\_d\_events(client\_number,id,name,event\_date,description,cost,venue\_id,package\_code,theme\_code)  
VALUES(7125,140, 'Cline Bas Mitzvah',  
TO\_DATE('15-Jul-2004','dd-Mon-yyyy'), 'Church and Private Home formal', 4500,105,87,77);
- Explanation: A new row in inserted into the table
- 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
  - ALTER TABLE copy\_d\_clients  
DISABLE CONSTRAINT copy\_d\_clients\_pk;
  - Explanation: The primary key constraints are temporarily disabled, which permit changes that would usually violate the constraints on primary keys
- 8. Repeat question 6: Insert the new values in 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');
  - Explanation: The primary key constraints are temporarily disabled and there is no longer a uniqueness restriction
- 9. Enable the primary-key constraint in the copy\_d\_clients table. Explain your results
  - ALTER TABLE copy\_d\_clients  
ENABLE CONSTRAINT copy\_d\_clients\_pk;
  - Explanation: The primary key being enabled will restore the uniqueness rules and restrictions
- 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?
  - Constraints may be disabled and reenabled to allow for updates and modifications while also ensuring that data integrity is maintained after the changes

- 12. Query the data dictionary for some of the constraints that you have created.  
How does the data dictionary identify each constraint type
  - `SELECT constraint_name, constraint_type, table_name`  
`FROM user_constraints`  
`WHERE table_name IN ('COPY_D_CLIENTS', 'COPY_D_EVENTS');`
  - **C**: Check constraint  
**P**: Primary key constraint  
**U**: Unique constraint  
**R**: Referential integrity (Foreign key constraint)  
**V**: View check constraint  
**O**: Other (e.g., domain constraints)

### Database Programming with SQL 15-1: Creating Views Practice Activities

- Vocabulary: Identify the vocabulary word for each definition below
  - A subset of data one or more tables that is generated from a query and stored as a virtual table
    - Answer: View
  - Name of view
    - Answer: View Name
  - Creates a view regardless of whether or not the base tables exist
    - Answer: Create force view
  - Derives data from a table , no functions or groups, performs DML operations through the view
    - Answer: Simple view
  - Create the view only if the base table exists
    - Answer: Create noforce view
  - Statement used to create a new view
    - Answer: Create view
  - Specifies a name for each expression selected by the view's query
    - Answer: Column alias
  - A complete SELECT statement
    - Answer: 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
    - Answer: Complex view
  - Re-create the view if it already exists
    - Answer: Create or replace view
- Try It / Solve It
  - 1. What are three uses for a view from a DBA's perspective?



- The three uses are for data security, simplifying queries, and data abstraction
- 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
  - CREATE OR REPLACE VIEW view\_d\_songs AS
 

```
SELECT
    id,
    title AS "Song Title",
    artist
FROM
    djs_on_demand
WHERE
    type_code = 'New Age';
```
- 3. SELECT \*FROM view\_d\_songs. What was returned?
  - Id, Song title, and 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
    id AS "Song ID",
    title AS "Song Title",
    artist AS "Artist Name",
    type_code AS "Type Code"
FROM
    djs_on_demand
WHERE
    type_code = '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

- 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
    - Answer: rownum
  - Specifies that insert and update performed through the view can't create rows which the view cannot select
    - Answer: With check option
  - Ensures that no DML operations can be performed on this view
    - Answer: With read only
- Try It / Solve It
  - 1. Query the data dictionary USER\_UPDATABLE\_COLUMNS to make sure the columns in the base tables will allow updating, 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

| ID | TITLE       | DURATION | ARTIST   | TYPE_CODE |
|----|-------------|----------|----------|-----------|
| 88 | Mello Jello | 2        | The What | 4         |

- INSERT INTO view\_copy\_d\_songs (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);
- 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 FROM read\_copy\_d\_cds WHERE cd\_number = 90;
- 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?

- `SELECT * FROM COPY_D_CDS AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '1' MINUTE);`
- 11. What are the restrictions on modifying data through a view?
  - Restrictions occur when the view involves multiple tables, aggregations, or complex expressions as well as when tables lack primary keys or contains calculated columns or subqueries
- 12. What is Moore's Law?
  - The principle that computing power can be expected to double every two years
- 13. What is the "singularity" in terms of computing?
  - It refers to the hypothetical future when and where artificial intelligence will surpass human intelligence

### Database Programming with SQL 15-3: Managing Views Practice Activities

- Vocabulary: Identify the vocabulary word for each definition below
  - Asks for the N largest or smallest values in a column
    - Answer: USER
  - Removes a view
    - Answer: transaction
  - Subquery with an alias that can be used within a SQL statement
    - Answer: explicit
- 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
    - `CREATE VIEW view_copy_d_songs AS  
SELECT title, artist  
FROM copy_d_songs;`
    - `SELECT * FROM view_copy_d_songs;`
  - 2. Issue a `DROP view_copy_d_songs`. Execute a `SELECT *` statement to verify that the view has been deleted
    - `DROP VIEW view_copy_d_songs;`
    - `SELECT * FROM view_copy_d_songs;`
  - 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
    - `SELECT last_name, salary, RANK() OVER (ORDER BY salary DESC)  
AS rank  
FROM employees  
WHERE RANK() OVER (ORDER BY salary DESC) <= 3;`

- 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
  - SELECT e.last\_name, e.salary, e.department\_id, d.max\_salary  
FROM employees e  
JOIN (  
SELECT department\_id, MAX(salary) AS max\_salary  
FROM employees  
GROUP BY department\_id  
) d  
ON e.department\_id = d.department\_id;
- 5. Create a query that will return the staff members of Global Fast Foods ranked by salary from lowest to highest
  - SELECT staff\_name, salary, RANK() OVER (ORDER BY salary ASC)  
AS rank  
FROM global\_fast\_foods\_staff;

● 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
  - CREATE TABLE my\_departments AS  
SELECT \*  
FROM departments;
  - SELECT \* FROM my\_departments;
- 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.
  - DESCRIBE my\_departments;  
ALTER TABLE my\_departments DISABLE ALL CONSTRAINTS;
- 3. Create a view called view\_my\_departments that includes: department\_id and department\_name. CREATE VIEW view\_my\_departments AS SELECT department\_id, department\_name FROM my\_departments;
- 4. Add the following data to the my\_departments table using view\_my\_departments.

| department_id | department_name |
|---------------|-----------------|
| 105           | Advertising     |
| 120           | Custodial       |
| 130           | Planning        |

- INSERT INTO view\_my\_departments (department\_id, department\_name)  
VALUES (105, 'Advertising'),

(120, 'Custodial'),  
(130, 'Planning');

- 5. Create or enable the department\_id column as the primary key
  - ALTER TABLE my\_departments ADD CONSTRAINT pk\_department\_id PRIMARY KEY (department\_id);
- 6. Enter a new department named Human Resources into the my\_departments table using view\_my\_departments. Do not add a new department ID
  - INSERT INTO view\_my\_departments (department\_name) VALUES ('Human Resources');
- 7. Add the Human Resources department, department ID 220, to my\_departments using view\_my\_departments
  - INSERT INTO view\_my\_departments (department\_id, department\_name) VALUES (220, 'Human Resources');
- 8. Verify that the new additions to my\_departments were added using view\_my\_departments
  - SELECT \*  
FROM view\_my\_departments;
- 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
  - DROP VIEW view\_my\_departments;
  - CREATE VIEW view\_my\_departments AS  
SELECT department\_id, department\_name, location\_id  
FROM my\_departments;
  - SELECT \* FROM view\_my\_departments;
- 10. Make location\_id a NOT NULL column in the my\_departments table
  - ALTER TABLE my\_departments  
MODIFY location\_id NOT NULL;
  - DESCRIBE my\_departments;
- 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
  - CREATE VIEW us\_departments\_view AS  
SELECT d.department\_name,  
l.street\_address,  
l.city,  
l.state\_province  
FROM departments d  
JOIN locations l

```
ON d.location_id = l.location_id
WHERE l.country_id = 'US'; -- Assuming 'US' is the country_id for the
United States SELECT * FROM us_departments_view;
```

## Database Programming with SQL 16-1: Working with Sequences Practice Activities

- Vocabulary: Identify the vocabulary word for each definition below
  - Command that automatically generates sequential numbers
    - Answer: CREATE SEQUENCE
  - Generates a numeric value
    - Answer: NEXTVAL
  - Returns the next available sequence value
    - Answer: NEXTVAL
  - Specifies the interval between sequence numbers
    - Answer: INCREMENT BY
  - Specifies a maximum value of  $10^{27}$  for an ascending sequence and -1 for a descending sequence (default)
    - Answer: MAXVALUE
  - 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
    - Answer: CURRVAL
  - Specifies a minimum value of 1 for an ascending sequence and  $-(10^{26})$  for a descending sequence (default)
    - Answer: MINVALUE (default)
  - Specifies a maximum or default value the sequence can generate
    - Answer: MAXVALUE (default)
  - Specifies the first sequence number to be generated
    - Answer: START WITH
  - Specifies how many values the Server pre-allocates and keeps in memory
    - Answer: CACHE
- 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
    - CREATE TABLE seq\_d\_songs AS  
SELECT \*  
FROM d\_songs;
  - 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.

- CREATE SEQUENCE seq\_d\_songs\_seq  
START WITH 100  
MAXVALUE 1000  
INCREMENT BY 2  
NOCACHE  
NOCYCLE;
- 3. Query the USER\_SEQUENCES data dictionary to verify the seq\_d\_songs\_seq SEQUENCE settings
  - SELECT sequence\_name, min\_value, max\_value, increment\_by,  
cycle\_flag, cache\_size  
FROM user\_sequences  
WHERE sequence\_name = 'SEQ\_D\_SONGS\_SEQ';
- 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

| ID | TITLE            | DURATION | ARTIST             | TYPE_CODE |
|----|------------------|----------|--------------------|-----------|
|    | Island Fever     | 5 min    | Hawaiian Islanders | 12        |
|    | Castle of Dreams | 4 min    | The Wanderers      | 77        |

- INSERT INTO seq\_d\_songs (id, title, duration, artist, type\_code)  
VALUES (seq\_d\_songs\_seq.NEXTVAL, 'Island Fever', '5 min', 'Hawaiian  
Islanders', 12);
- INSERT INTO seq\_d\_songs (id, title, duration, artist, type\_code)  
VALUES (seq\_d\_songs\_seq.NEXTVAL, 'Castle of Dreams', '4 min', 'The  
Wanderers', 77);
- 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)
  - SELECT seq\_d\_songs\_seq.CURRVAL
  - FROM DUAL;
- 6. What are three benefits of using SEQUENCES?
  - The benefits are the automated creation of unique identifiers, improved performance, and controlled concurrent access
- 7. What are the advantages of caching sequence values?
  - The advantages are improved performance, efficient resource usage, and reduced contention
- 8. Name three reasons why gaps may occur in a sequence?
  - They may occur due to transaction rollbacks, system crashes, and manual adjustments
- Extension Exercises



- 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)
  - CREATE TABLE students (
   
          student\_id NUMBER(10),
   
          first\_name VARCHAR2(50),
   
          last\_name VARCHAR2(50),
   
          birth\_date DATE,
   
          major VARCHAR2(100)
   
);
- 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
  - CREATE SEQUENCE student\_id\_seq
   
          START WITH 1
   
          INCREMENT BY 1
   
          MAXVALUE 1000000
   
          NOCACHE
   
          NOCYCLE;
- 3. Now write the code to add students to your STUDENTS table, using your sequence “database object”
  - INSERT INTO students (student\_id, first\_name, last\_name, birth\_date, major)
   
VALUES (student\_id\_seq.NEXTVAL, 'John', 'Doe',
   
TO\_DATE('2000-01-15', 'YYYY-MM-DD'), 'Computer Science');
  - INSERT INTO students (student\_id, first\_name, last\_name, birth\_date, major)
   
VALUES (student\_id\_seq.NEXTVAL, 'Jane', 'Smith',
   
TO\_DATE('1999-03-22', 'YYYY-MM-DD'), 'Biology');

## Database Programming with SQL 16-2: Indexes and Synonyms Practice Activities

- Vocabulary: Identify the vocabulary word for each definition below
  - Confirms the existence of indexes from the USER\_INDEXES data dictionary view
    - Answer: USER\_INDEXES
  - Schema object that speeds up retrieval of rows
    - Answer: INDEX
  - To refer to a table by another name to simplify access

- Answer: ALIAS
- An index that you create on multiple columns in a table
  - Answer: COMPOSITE INDEX
- 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
  - Answer: UNIQUE INDEX
- Stores the indexed values and uses the index based on a SELECT statement to retrieve the data
  - Answer: B-TREE INDEX
- Removes an index
  - Answer: DROP INDEX
- Gives alternative names to objects
  - Answer: SYNONYM
- Try It / Solve It
  - 1. What is an index and what is it used for?
    - An index is a schema object in a database. It is used to improve the speed of querying data and avoids full table scans
  - 2. What is a ROWID, and how is it used?
    - ROWID is a unique identifier and references the location of a row in the database. It is used to retrieve rows and often used through indexes
  - 3. When will an index be created automatically?
    - It is created automatically when the primary key and unique key are defined
  - 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
    - CREATE INDEX idx\_cd\_number ON d\_track\_listings(cd\_number);
    - SELECT index\_name, table\_name, uniqueness  
FROM user\_indexes  
WHERE table\_name = 'D\_TRACK\_LISTINGS';
  - 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
    - SELECT ui.index\_name, ui.table\_name, ui.uniqueness  
FROM user\_indexes ui  
JOIN user\_ind\_columns uic  
ON ui.index\_name = uic.index\_name  
WHERE ui.table\_name = 'D\_SONGS';
  - 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

- SELECT index\_name, table\_name, uniqueness  
FROM user\_indexes  
WHERE table\_name = 'D\_EVENTS';
- 7. Write a query to create a synonym called dj\_tracks for the DJs on Demand d\_track\_listings table
  - CREATE SYNONYM dj\_tracks  
FOR d\_track\_listings;
- 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.
  - CREATE INDEX idx\_last\_name\_func  
ON d\_partners(LOWER(last\_name));
  - SELECT \* FROM d\_partners WHERE LOWER(last\_name) = 'smith';
- 9. Create a synonym for the D\_TRACK\_LISTINGS table. Confirm that it has been created by querying the data dictionary
  - SELECT synonym\_name, table\_owner, table\_name  
FROM user\_synonyms  
WHERE synonym\_name = 'DJ\_TRACKS';
- 10. Drop the synonym that you created in question 9
  - DROP SYNONYM dj\_tracks;

## Database Programming with SQL 17-1: Controlling User Access Practice Activities

### ● Try It / Solve It

- 1. What are system privileges concerned with?
  - They are concerned with the ability to access the database on a system level and perform tasks
- 2. What are object privileges concerned with?
  - They are concerned with the ability to access and perform actions on database objects
- 3. What is another name for object security?
  - Another name is data security
- 4. What commands are necessary to allow Scott access to the database with a password of tiger?
  - CREATE USER Scott  
IDENTIFIED BY tiger;
- 5. What are the commands to allow Scott to SELECT from and UPDATE the d\_clients table?
  - GRANT SELECT, UPDATE ON d\_clients TO Scott;
- 6. What is the command to allow everybody the ability to view the d\_songs table?
  - GRANT SELECT ON d\_songs TO PUBLIC;

- 7. Query the data dictionary to view the object privileges granted to you the user
  - `SELECT * FROM USER_TAB_PRIVS_RECD;`
- 8. What privilege should a user be given to create tables?
  - `GRANT CREATE TABLE TO username;`
- 9. If you create a table, how can you pass along privileges to other users just to view your table?
  - `GRANT SELECT ON new_table TO username;`
- 10. What syntax would you use to grant another user access to your copy\_employees table?
  - `GRANT SELECT, INSERT, UPDATE, DELETE ON copy_employees TO other_user;`
- 11. How can you find out what privileges you have been granted for columns in the tables belonging to others?
  - `SELECT * FROM USER_COL_PRIVS_RECD WHERE GRANTEE = your_username;`

## Database Programming with SQL 17-2: Creating and Revoking Object Privileges Practice Activities

- Try It / Solve It

- 1. What is a role?
  - It is a group of privileges that can be granted to users
- 2. What are the advantages of a role to a DBA?
  - They simplify the process of maintaining and revoking privileges
- 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.
  - `GRANT SELECT ON new_table TO other_user WITH GRANT OPTION;`
- 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?
  - `CREATE ROLE role_name;`
  - `GRANT CREATE SESSION, CREATE TABLE, SELECT ANY TABLE TO role_name;`
  - `GRANT role_name TO user1, user2, user3;`
- 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
    - `CREATE ROLE manager;`
    - `GRANT SELECT, INSERT, UPDATE, DELETE ON employees TO manager;`

- b. Create a role of clerk that just has the privileges of select and insert on the employees table
  - CREATE ROLE clerk;
  - GRANT SELECT, INSERT ON employees TO clerk;
- c. Grant the manager role to user scott
  - GRANT manager TO scott;
- d. Revoke the ability to delete from the employees table from the manager role
  - REVOKE DELETE ON employees FROM manager;
- 6. What is the purpose of a database link?
  - It allows one database to communicate with and access objects in another database

### Database Programming with SQL 17-3: Regular Expressions Practice Activities

- 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)
    - SELECT \*  
FROM employees  
WHERE REGEXP\_LIKE(first\_name, '^S[t|h]');
  - 2. Investigate the LOCATIONS table
    - a. Describe the table
      - DESC LOCATIONS;
    - b. Perform a select that returns all rows and all columns of that table
      - SELECT \*  
FROM LOCATIONS;
    - c. Write a query using regular expressions that removes the spaces in the street\_address column in the LOCATIONS table
      - SELECT REGEXP\_REPLACE(street\_address, '\s+', '') AS  
street\_address\_no\_spaces  
FROM LOCATIONS;