

- **14-1 Intro to Constraints; NOT NULL and UNIQUE Constraints**

- *Vocabulary*

- 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 Constraint
- 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 Constraint
- References one or more columns and is defined separately from the definitions of the columns in the table
 - Out-of-line Constraint
- Database rule
 - Constraint
- Database rule that references a single column
 - Column-level Constraint

- *Try it/ Solve it*

- 1. What is a “constraint” as it relates to data integrity?
 - Constraints are any rules or conditions that need to be followed in a table to ensure data integrity.
- 2. What are the limitations of constraints that may be applied at the column level and at the table level?
 - Column-level constraint limitations only apply to a single column and are defined right after the column name. Whereas with Table-Level constraints limitations are applied after defining a column and has the ability to enforce complex or many columns.

- 3. Why is it important to give meaningful names to constraints?
 - Meaningful name convention for constraints is important to ensure that there is a clear understanding of the purpose of interoperability. Other than just clarity, it also makes it maintainable as database constraints adapt over time.
- 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.
 - VARCHAR2: For variable-length character data
 - CHAR: fixed-length character data
 - NUMBER
 - DATE
- 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, -- This column
    cannot be NULL
    last_name VARCHAR2(50), -- This column can be NULL by
    default
    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, -- Primary key on location_id
    location_name VARCHAR2(100) NOT NULL, -- Location
    name cannot be NULL
    address VARCHAR2(255), -- Address is nullable by default
    city VARCHAR2(50) NOT NULL, -- City is mandatory
    postal_code VARCHAR2(20), -- Postal code is nullable
    country VARCHAR2(50) NOT NULL, -- Country is mandatory
    latitude NUMBER(9, 6), -- Latitude (nullable)
    longitude NUMBER(9, 6), -- Longitude (nullable)
    created_date DATE DEFAULT SYSDATE NOT NULL -- Date
    when the location was created (defaults to current date)
);
```

- 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 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 locations (  
    location_id NUMBER(6) CONSTRAINT pk_location_id  
PRIMARY KEY, -- Primary key constraint on location_id  
    location_name VARCHAR2(100) NOT NULL, -- Location  
name cannot be NULL  
    address VARCHAR2(255), -- Address is nullable by default  
    city VARCHAR2(50) NOT NULL, -- City is mandatory  
    postal_code VARCHAR2(20), -- Postal code is nullable  
    country VARCHAR2(50) NOT NULL, -- Country is mandatory  
    latitude NUMBER(9, 6), -- Latitude (nullable)  
    longitude NUMBER(9, 6), -- Longitude (nullable)  
    created_date DATE DEFAULT SYSDATE NOT NULL, -- Date  
when the location was created (defaults to current date)  
  
    -- UNIQUE constraints defined at the table level
```

```

        CONSTRAINT uq_location_name UNIQUE (location_name),
-- Ensures unique location names
        CONSTRAINT uq_postal_code UNIQUE (postal_code) --
Ensures unique postal codes
);

```

- **14-2 PRIMARY KEY, FOREIGN KEY, and CHECK Constraints**

- Vocabulary

- 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 NULL
- 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 Constraint

- Try it/ Solve it

- 1.What is the purpose of a
 - a.PRIMARY KEY
 - Provides a unique identifier
 - b.FOREIGN KEY
 - Establishes a relationship between two tables
 - c.CHECK CONSTRAINT
 - Ensures 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.
 animal_id NUMBER(6) name VARCHAR2(25) license_tag_number NUMBER(10) admit_date DATE adoption_id NUMBER(5), vaccination_date DATE

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.

```
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

- It means that the parent tables is deleted and all the rows from animals table are also deleted
- b. ON DELETE SET NULL
 - It means that the parent table is deleted but the animal table will still have key to parent table is now set to null
- 7.What are the restrictions on defining a CHECK constraint?
Conditions for check constraints must have a true or false value.

● 14-3 Managing Constraints

- *Vocabulary*
 - To deactivate an integrity constraint
 - Disable Constraint
 - Disables dependent integrity constraints
 - CASCADE constraint disable
 - 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
 - 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 newtables; only the column datatype definitions remain. You will need to add a PRIMARY KEYconstraint 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 tablename 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;
```

Explain Results :

By using the key constraint name copy_d_clients, the result of this action is that the client number column no longer has a primary key enforcement. Basically this means that there are no duplicates in the columns.

- 6. 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;
```

Explain results:

Disable constraint temporarily disables primary key constraints which allows you to create changes to the table that would typically violate the constraints in place for the primary key values.

- 7. 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');
```

Explain results:

When you temporarily remove the PK constraints, which means that there's not enforce uniqueness. This can lead to referential integrity risks

or issues because events and client numbers values will not match a valid client number.

- 8. Enable the primary-key constraint in the copy_d_clients table.
ALTER TABLE copy_d_clients
ENABLE CONSTRAINT copy_d_clients_pk;
 - 9. 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;
 - 10. Why might you want to disable and then re-enable a constraint?
The reason that you might want to disable and re-enable specific constraints is to maintain referential integrity, improve performance of the database, evolve schema changes in infrastructure, and be able to resolve any data issues.
 - 11. 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)

- **15-1 Creating Views**

- *Vocabulary*

- A subset of data 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 force view
 - Derives data from a table, no functions or groups, performs DML operations through the view

- Simple view
 - Create the view only if the base table exists
 - Create noforce 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-create the view if it already exists
 - Create or replace view
- Try it/ Solve it
- 1.What are three uses for a view from a DBA's perspective?
Data Security , simplification, 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;

```

- **15-2 DML Operations and Views**

- *Vocabulary*

- A pseudocolumn which assigns a sequential value starting with 1 to each of the rows returned from the subquery
 - rownum
 - Specifies that insert and update performed through the view can't create rows which the view cannot select
 - With check option
 - Ensures that no DML operations can be performed on this view
 - 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.
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?
Read-only view is one type of restriction that won't allow you to perform INSERT, UPDATE, or DELETE functionalities. In addition, viewing aggregate functions/ clauses are not updatable which can lead to inconsistencies in data summary.

- 12. What is Moore's Law?
Moore's Law is the idea of our overall computing power to double every 2 years.
- 13. What is the "singularity" in terms of computing?
Often refer to when technology intelligence surpass that of 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;