

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

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

1. What is a “constraint” as it relates to data integrity?
 - a. 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?
 - a. Some of the limitations of column level constraints include limitations that only apply to a a single column and are defined right after the column name. 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?
 - a. It is important to give meaningful names to constraints because it ensures that there is a clear understanding of the purpose of interoperability. It 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
 - a. VARCHAR2: For variable-length character data
 - b. • CHAR: fixed-length character data
 - c. • NUMBER
 - d. • DATE
5. Use "nullable" to indicate those columns that can have null values
 - a.

```
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.
 - a.

```
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.
 - a.

```
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.
 - a. 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
 - a. 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

ON DELETE CASCADE	Allows a foreign key row that is referenced to a primary key row to be deleted
Check Constraint	Explicitly defines a condition that must be met

Primary Key	A column or set of columns that uniquely identifies each row in a table
Not NULL constraint	Constraint ensures that the column contains no null values
ON DELETE SET NULL	Allows a child row to remain in a table with null values when a parent record has been deleted
Foreign Key Constraint	Establishes a relationship between the foreign key column and a primary key or unique key in the same table or a different table

1. What is the purpose of a
 - a. PRIMARY KEY
 - i. Provides a unique identifier
 - b. FOREIGN KEY
 - i. Establishes a relationship between two tables
 - c. CHECK CONSTRAINT
 - i. 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

 - a. Table-Level Constraints:
 - i. PRIMARY KEY (animal_id)
 - ii. UNIQUE (license_tag_number)
 - b. Column-Level Constraints:
 - i. admit_date DATE NOT NULL
 - ii. vaccination_date DATE NOT NULL

3. Create the animals table. Write the syntax you will use to create the table.
 - a. 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
 - a. 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
 - i. It means that the parent tables is deleted and all the rows from animals table are also deleted
 - b. ON DELETE SET NULL
 - i. 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

Disable Constraint	To deactivate an integrity constraint
CASCADE constraint disable	Disables dependent integrity constraints
Alter Table	To add, modify, or drop columns from a table
Enable constraint	To activate an integrity constraint currently disabled
Drop Constraint	Removes a constraint from a table

Drop Column	Allows user to delete a column from a table
On delete/on update clause	Defines the actions the database server takes when a user attempts to delete or update a key to which existing foreign keys point

1. What are four functions that an ALTER statement can perform on constraints?
 - a. Constraint
 - b. Drop
 - c. Modify
 - d. 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?
 - a. ALTER TABLE copy_d_clients
 - b. ADD CONSTRAINT copy_d_clients_pk PRIMARY KEY
 - c. (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?
 - a. ALTER TABLE copy_d_events
 - b. ADD CONSTRAINT copy_d_events_fk FOREIGN KEY (client_number)
 - c. 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
 - a. ALTER TABLE copy_d_clients
 - b. DROP CONSTRAINT copy_d_clients_pk;
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.
 - a. ALTER TABLE copy_d_clients
 - b. DISABLE CONSTRAINT copy_d_clients_pk;
7. Repeat question 6: Insert the new values in the copy_d_events table

- a. INSERT INTO copy_d_events (client_number, event_date, event_details)
VALUES (12345, TO_DATE('2024-11-22', 'YYYY-MM-DD'), 'Event 2');
8. Enable the primary-key constraint in the copy_d_clients table.
 - a. ALTER TABLE copy_d_clients
 - b. 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?
 - a. ALTER TABLE copy_d_events
 - b. ENABLE CONSTRAINT copy_d_events_fk;
10. Why might you want to disable and then re-enable a constraint?
 - a. 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
 - a. SELECT constraint_name, constraint_type, table_name
 - b. FROM user_constraints
 - c. WHERE table_name IN ('COPY_D_CLIENTS', 'COPY_D_EVENTS');
 - d. C: Check constraint
 - e. P: Primary key constraint
 - f. U: Unique constraint
 - g. R: Referential integrity (Foreign key constraint)
 - h. V: View check constraint
 - i. O: Other (e.g., domain constraints)

15-1 Creating Views

View	A subset of data one or more tables that is generated from a query and stored as a virtual table
View Name	Name of view
Create force view	Creates a view regardless of whether or not the base tables exist
Simple view	Derives data from a table , no functions or groups, performs DML operations through the view
Create noforce view	Create the view only if the base table exists
Create view	Statement used to create a new view

Column alias	Specifies a name for each expression selected by the view's query
View query	A complete SELECT statement
Complex view	Derives data from more than one table , contains functions or groups of data, and does not always allow DML operations through the view
Create or replace view	Re-create the view if it already exists

1. What are three uses for a view from a DBA's perspective?
 - a. Data Security
 - b. Simplification
 - c. 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 alia "Song Title" for the title column.
 - a.

```
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?
 - a. Id, Song title, and Artist
4. REPLACE view_d_songs. Add type_code to the column list. Use aliases for all columns
 - a.

```
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

- a.

```
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

rownum	A pseudocolumn which assigns a sequential value starting with 1 to each of the rows returned from the subquery
With check option	Specifies that insert and update performed through the view can't create rows which the view cannot select
With read only	Ensures that no DML operations can be performed on this view

1.

```
SELECT TABLE_NAME, COLUMN_NAME, UPDATABLE
FROM USER_UPDATABLE_COLUMNS
WHERE UPDATABLE = 'YES';
```
2.

```
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.
 - a. INSERT INTO view_copy_d_songs (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
 4. CREATE OR REPLACE VIEW read_copy_d_cds AS
SELECT *
FROM COPY_D_CDS
WHERE year = 2000
WITH READ ONLY;
 5. DELETE FROM read_copy_d_cds WHERE cd_number = 90;
 6. 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. DELETE FROM read_copy_d_cds
WHERE year = 2000;
 8. DELETE FROM read_copy_d_cds
WHERE cd_number = 90;
 9. DELETE FROM read_copy_d_cds
WHERE year = 2001;
 10. SELECT * FROM COPY_D_CDS AS OF TIMESTAMP
(SYSTIMESTAMP - INTERVAL '1' MINUTE);
 11. 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. Moore's Law is the idea of our overall computing power to double every 2 years.
 13. Singularity = when technology intelligence surpass that of human intelligence

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

USER	Asks for the N largest or smallest values in a column
transaction	Removes a view
explicit	Subquery with an alias that can be used within a SQL statement

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
 - a. CREATE VIEW view_copy_d_songs AS
 - b. SELECT title, artist
 - c. FROM copy_d_songs;
 - d. 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
 - a. DROP VIEW view_copy_d_songs;
 - b. SELECT * FROM view_copy_d_songs;
 - c.
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
 - a. 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
 - a. 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
 - a. SELECT staff_name, salary, RANK() OVER (ORDER BY salary ASC)
AS rank
FROM global_fast_foods_staff;

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

CREATE SEQUENCE	Command that automatically generates sequential numbers
NEXTVAL	Generates a numeric value
NEXTVAL	Returns the next available sequence value
INCREMENT BY	Specifies the interval between sequence numbers
MAXVALUE	Specifies a maximum value of 10^{27} for an ascending sequence and -1 for a descending sequence (default)
CURRVAL	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
MINVALUE (default)	Specifies a minimum value of 1 for an ascending sequence and $-(10^{26})$ for a descending sequence (default)
MAXVALUE (default)	Specifies a maximum or default value the sequence can generate
START WITH	Specifies the first sequence number to be generated
CACHE	Specifies how many values the Server pre-allocates and keeps in memory

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
 - a. 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.

- a. 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
 - a. 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
 - a. INSERT INTO seq_d_songs (id, title, duration, artist, type_code)
VALUES (seq_d_songs_seq.NEXTVAL, 'Island Fever', '5 min', 'Hawaiian Islanders', 12);
 - b. 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)
 - a. SELECT seq_d_songs_seq.CURRVAL
 - b. FROM DUAL;
6. What are three benefits of using SEQUENCES?
 - a. The benefits are the automated creation of unique identifiers, improved performance, and controlled concurrent access
7. What are the advantages of caching sequence values?
 - a. The advantages are improved performance, efficient resource usage, and reduced contention
8. Name three reasons why gaps may occur in a sequence?
 - a. They may occur due to transaction rollbacks, system crashes, and manual Adjustments

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

USER_INDEXES	Confirms the existence of indexes from the
--------------	--

	USER_INDEXES data dictionary view
INDEX	Schema object that speeds up retrieval of rows
ALIAS	To refer to a table by another name to simplify access
COMPOSITE INDEX	An index that you create on multiple columns in a table
UNIQUE 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
B-TREE INDEX	Stores the indexed values and uses the index based on a SELECT statement to retrieve the data
DROP INDEX	Removes an index
SYNONYM	Gives alternative names to objects

1. What is an index and what is it used for?
 - a. 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?
 - a. 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?
 - a. 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
 - a. CREATE INDEX idx_cd_number ON d_track_listings(cd_number);
 - b. 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
 - a. 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
 - a. SELECT index_name, table_name, uniqueness
 - b. FROM user_indexes
 - c. 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
 - a. CREATE SYNONYM dj_tracks
 - b. 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.
 - a. 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
 - a. 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
 - a. DROP SYNONYM dj_tracks;

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

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

7. Query the data dictionary to view the object privileges granted to you the user
 - a. `SELECT * FROM USER_TAB_PRIVS_RECD;`
8. What privilege should a user be given to create tables?
 - a. `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?
 - a. `GRANT SELECT ON new_table TO username;`
10. What syntax would you use to grant another user access to your copy_employees table?
 - a. `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?
 - a. `SELECT * FROM USER_COL_PRIVS_RECD WHERE GRANTEE = your_username;`

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

1. What is a role?
 - a. It is a group of privileges that can be granted to users
2. What are the advantages of a role to a DBA?
 - a. 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.
 - a. `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?
 - a. `CREATE ROLE role_name;`
 - b. `GRANT CREATE SESSION, CREATE TABLE, SELECT ANY TABLE TO role_name;`
 - c. `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
 - i. `CREATE ROLE manager;`
 - ii. `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
 - i. `CREATE ROLE clerk;`
 - ii. `GRANT SELECT, INSERT ON employees TO clerk;`

- c. Grant the manager role to user scott
 - i. GRANT manager TO scott;
 - d. Revoke the ability to delete from the employees table from the manager Role
 - i. REVOKE DELETE ON employees FROM manager;
6. What is the purpose of a database link?
- a. It allows one database to communicate with and access objects in another database

Database Programming with SQL 17-3: Regular Expressions Practice Activitie

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)
 - a. SELECT *
FROM employees
WHERE REGEXP_LIKE(first_name, '^S[t|h]');
2. Investigate the LOCATIONS table
 - a. Describe the table
 - i. DESC LOCATIONS;
 - b. Perform a select that returns all rows and all columns of that table
 - i. SELECT *
FROM LOCATIONS;
 - c. Write a query using regular expressions that removes the spaces in the street_address column in the LOCATIONS table
 - i. SELECT REGEXP_REPLACE(street_address, '\s+', '') AS
street_address_no_spaces
FROM LOCATIONS;