Practice 14-1

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
PK	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
references	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 must be unique
FK	Designates a column (child table) that establishes a relationship between a primary key in the same table and a different table (prent table)
Table level constraint	References one or more columns and is defined separately from the definition 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? It maintains it and stops user from deleting certain data (rows/columns) when updating a table or inserting data.
2. What are the limitations of constraints that may be applied at the column level and at the table level?
   a. Column: they can't involve multiple columns so data that requires relationships between multiple columns can't be defined
   b. Table: can only involved that particular table or the table with the FK
3. Why is it important to give meaningful names to constraints? Improved readability and maintenance, easier debugging, and avoids conflicts so it is easier when migrating schemas or integrating multiple databases
4. Based on the information provided by the owners, choose a datatype for each column. Indicate the length, precision, and scale for each NUMBER datatype.
5. Use "nullable" to indicate those columns that can have null values.
   a. No, yes, no, no, no, yes, yes. Yes, yes.
6. Write the CREATE TABLE statement for the Global Fast Foods locations table to define the constraints at the column level.

CREATE TABLE f_global_locations

```
( id NUMBER(6,0) CONSTRAINT f_gln_id_pk PRIMARY KEY ,
name VARCHAR2(50),
date_opened DATE CONSTRAINT f_gln_dt_opened_nn NOT NULL ENABLE,
address VARCHAR2(50) CONSTRAINT f_gln_add_nn NOT NULL ENABLE,
city VARCHAR2(30)  CONSTRAINT f_gln_city_nn NOT NULL ENABLE,
zip_postal_code VARCHAR2(12),
phone VARCHAR2(20),
email VARCHAR2(75) CONSTRAINT f_gln_email_uk UNIQUE,
manager_id NUMBER(6,0),
emergency_contact VARCHAR2(20)
);
```

7. Execute the CREATE TABLE statement in Oracle Application Express.
8. Execute a DESCRIBE command to view the Table Summary information.
   `DESCRIBE f_global_locations;`
9. Rewrite the CREATE TABLE statement for the Global Fast Foods locations table to define the UNIQUE constraints at the table level. Do no execute this statement.

```
CREATE TABLE  f_global_locations
( id NUMBER(6,0) CONSTRAINT f_gln_id_pk PRIMARY KEY ,
name VARCHAR2(50),
date_opened DATE CONSTRAINT f_gln_dt_opened_nn NOT NULL ENABLE,
address VARCHAR2(50) CONSTRAINT f_gln_add_nn NOT NULL ENABLE,
city VARCHAR2(30)  CONSTRAINT f_gln_city_nn NOT NULL ENABLE,
zip_postal_code VARCHAR2(12),
phone VARCHAR2(20),
email VARCHAR2(75) ,
manager_id NUMBER(6,0),
emergency_contact VARCHAR2(20),
 CONSTRAINT f_gln_email_uk UNIQUE(email)

)
```

Practice 14-2

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
PK	A column or set of columns that uniquely identifies each row in a table
Not null	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
FK constraint	Establishes a relationship between the foreign key column

	primary key or unique key in the same table or a different table

1. What is the purpose of a
    a. Primary Key: uniquely identifies each record in a table
    b. Foreign Key:enforces relationships between two tables, ensure data integrity
    c. Check Constraint: enforces a condition that must be met
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
3. Create the animals table. Write the syntax you will use to create the table.

```
CREATE TABLE  animals
( animal_id  NUMBER(6,0) CONSTRAINT anl_anl_id_pk PRIMARY KEY ,
name VARCHAR2(25),
license_tag_number NUMBER(10,0) CONSTRAINT anl_l_tag_num_uk UNIQUE,
admit_date  DATE CONSTRAINT anl_adt_dat_nn NOT NULL ENABLE,
adoption_id   NUMBER(5,0),
vaccination_date  DATE CONSTRAINT anl_vcc_dat_nn NOT NULL ENABLE)
```

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

ANIMAL_ID	NAME	LICENSE_TAG_NUMBER	ADMIT_DATE	ADOPTION_ID	VACCINATION_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, 'Spot', 35540, TO_DATE('10-Oct-2004', 'DD-Mon-YYYY'), 205, TO_DATE('12-Oct-2004',
'DD-Mon-YYYY'));
SELECT * FROM animals;
```

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.

```
ALTER TABLE animals
MODIFY ( adoption_id   NUMBER(5,0) CONSTRAINT anl_adopt_id_fk  REFERENCES adoptions(id)
ENABLE )

SELECT *
FROM user_constraints
WHERE LOWER(table_name) = 'animals' AND constraint_type = 'R'

ALTER TABLE animals
DROP CONSTRAINT anl_adopt_id_fk

SELECT *
FROM user_constraints
WHERE LOWER(table_name) = 'animals' AND constraint_type = 'R';
```

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

```
ALTER TABLE  animals
ADD CONSTRAINT anl_adopt_id_fk  FOREIGN KEY (adoption_id)
        REFERENCES  adoptions (id) ENABLE ;

SELECT delete_rule
FROM user_constraints
WHERE LOWER(table_name) = 'animals' AND constraint_type = 'R';


ALTER TABLE animals
DROP CONSTRAINT anl_adopt_id_fk  ;
ALTER TABLE  animals
ADD CONSTRAINT anl_adopt_id_fk  FOREIGN KEY (adoption_id)
REFERENCES  adoptions(id) ON DELETE SET NULL  ENABLE ;
SELECT delete_rule
FROM user_constraints
WHERE LOWER(table_name) = 'animals' AND constraint_type = 'R';

DELETE FROM adoptions WHERE id= 500;

SELECT * FROM animals;
```

7. What are the restrictions on defining a CHECK constraint? Must be a valid boolean expresssion

Practice 14-3

Disable constraint	To deactivate an integrity constraint
Cascade clause	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
Cascade constraints	Defines the actions the database server takes when a user attempts to delete or update a key to which existing foreign keys point

Using Oracle Application Express, click the SQL Workshop tab in the menu bar. Click the Object Browser and verify that you have a table named copy_d_clients and a table named copy_d_events. If you don't have these tables in your schema, create them before completing the exercises below. Here is how the original tables are related. The d_clients table has a primary key client_number. Thishas a primary-key constraint and it is referenced in the foreign-key constraint on the d_events table

1. What are four functions that an ALTER statement can perform on constraints? Add, drop, modify, enable or disable constraints
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_clt_client_number_pk PRIMARY KEY (client_number);

SELECT *
FROM user_constraints
WHERE LOWER(table_name) =  'copy_d_clients' and constraint_type = 'P';
```

3. Create a FOREIGN KEY constraint in the copy_d_events table. Name the foreign key copy_d_events_fk. This key references the copy_d_clients table client_number column. What is the syntax you used to create the FOREIGN KEY constraint in the copy_d_events table?

```
ALTER TABLE  copy_d_events
ADD CONSTRAINT copy_d_eve_client_number_fk FOREIGN KEY (client_number) REFERENCES
copy_d_clients (client_number) ENABLE;

SELECT *
```

FROM user_constraints
WHERE LOWER(table_name) = 'copy_d_events' and constraint_type = 'R';

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 constraints name for the primary key in the copy_d_clients table is_____.
       COPY_D_CLT_CLIENT_NUMBER_PK
    b. The constraint name for the foreign key in the copy_d_event table is _____.
       COPY_D_EVE_CLIENT_NUMBER_FK

5. Drop the PRIMARY KEY constraint on the copy_d_clients table. Explain your result.
   ORA-02273: this unique/primary key is referenced by some foreign keys
6. Add the following event to the copy_d_events table. Explain your result.
   ORA-02291: integrity constraint (HKUMAR.COPY_D_EVE_CLIENT_NUMBER_FK) violated - parent key not found
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.
   values from #5 to the copy_d_events table. Explain your results.
   ALTER TABLE copy_d_clients
   DISABLE CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK ;
   ORA-02297: cannot disable constraint (HKUMAR.COPY_D_CLT_CLIENT_NUMBER_PK) - dependencies exist

   ALTER TABLE copy_d_clients
   DISABLE CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK CASCADE;
   Table altered

8. Repeat question6: Insert the new values in the copy_d_events table. Explain your results.
   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);
   1 row(s) inserted.

9. Enable the primary key constraint in the copy_d_clients table. Explain your results.

   ALTER TABLE copy_d_clients
   ENABLE CONSTRAINT COPY_D_CLT_CLIENT_NUMBER_PK ;

10. If you wanted to enable the foreign-key column and reestablish the referential integrity between these two tables, what must be done?
    Fix row with client_number to a valid value or null
11. Why might you want to disable and then re-enable a constraint? Optimize bulk operations

12. Query the data dictionary for some of the constraints that you have created. How does the data dictionary identify each constraint type?
    C - check constraint, P - PK, R- FK, and U - unique key

Practice 15-1

view	A subset of data from one or more tables that is generated from a query and stored as a virtual table.
view_name	Name of view
force	Creates a view regardless of whether or not the base tables exist
Simple view	Derived data from a table, no functions or groups, performs DML operations through the view
noforce	Creates the view only if the base table exists
Create view statement	Statement used to create a new view
alias	Specifies a name for each expression selected by the view's query
subquery	A complete SELECT statement
Complex view	Derived data from more than one table, contains functions or groups of data, and does not always allow DML operations through the view
replace	Re-creates the view if it already exists

1. What are three uses for a view from a DBA's perspective?
   Data security, access control, simplify complex queries
2. Create a simple view called view_d_songs that contains the ID, title, and artist from the DJs on Demand table for each "New Age" type code. In the subquery, use the alias "Song Title" for the title column.

```
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';


CREATE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
```

```
where d_types.description = 'New Age';

CREATE OR REPLACE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';
```

Verify results again:

```
SELECT * FROM view_d_songs ;
```

3. SELECT *
   FROM view_d_songs.

   What was returned?

```
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';

SELECT * FROM view_d_songs ;
```

4. REPLACE view_d_songs. Add type_code to the column list. Use aliases for all columns.

```
CREATE OR REPLACE VIEW view_d_songs AS
SELECT d_songs.id, d_songs.title "Song Title", d_songs.artist, d_songs.type_code
from d_songs INNER JOIN d_types ON d_songs.type_code = d_types.code
where d_types.description = 'New Age';
```

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

```
CREATE OR REPLACE VIEW view_d_events_pkgs AS
SELECT evt.name "Name of Event", TO_CHAR(evt.event_date, 'dd-Month-yyyy')  "Event date",
thm.description "Theme description"
FROM  d_events  evt INNER JOIN d_themes  thm  ON evt.theme_code = thm.code
WHERE evt.event_date <= ADD_MONTHS(SYSDATE,1);
SELECT * FROM view_d_events_pkgs ;
```

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

```
DESCRIBE employees;

SELECT department_id FROM departments WHERE department_id NOT IN ( SELECT
NVL(department_id,0) FROM employees);
 Suggests:
CREATE OR REPLACE VIEW view_min_max_avg_dpt_salary ("Department Id", "Department Name",
"Max Salary", "Min Salary", "Average Salary") AS
SELECT dpt.department_id, dpt.department_name, MAX(NVL(emp.salary,0)), MIN(NVL(emp.salary,0)),
ROUND(AVG(NVL(emp.salary,0)),2)
FROM departments dpt LEFT OUTER JOIN employees emp ON dpt.department_id = emp.department_id
GROUP BY (dpt.department_id, dpt.department_name);

SELECT * FROM view_min_max_avg_dpt_salary;
```

Practice 15-2

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 INSERTS and UPDATES 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

Use the DESCRIBE statement to verify that you have tables named copy_d_songs, copy_d_events, copy_d_cds, and copy_d_clients in your schema. If you don't, write a query to create a copy of each.

1. Query the data dictionary USER_UPDATABLE_COLUMNS to make sure the columns in the base tables will allow UPDATE, INSERT, or DELETE. Use a select statement. All table name in the data dictionary are stored in uppercase.

   USER_UPDATABLE_COLUMNS describes columns in a join view that can be updated by the current user, subject to appropriate privileges.

   ```
   SELECT owner, table_name, column_name, updatable,insertable, deletable
   FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_songs';
   ```

```
SELECT owner, table_name, column_name, updatable,insertable, deletable
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_events';
```

```
SELECT owner, table_name, column_name, updatable,insertable, deletable
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_cds';
```

```
SELECT owner, table_name, column_name, updatable,insertable, deletable
FROM user_updatable_columns WHERE LOWER(table_name) = 'copy_d_clients';
```

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;

SELECT * FROM view_copy_d_songs;
```

3. Use view_copy_d_songs to INSERT the following fata into the underlying copy_d_songs table. Execute a SELECT * from copy_d_songs to verify your DML command.

```
        INSERT INTO view_copy_d_songs(id,title,duration,artist,type_code)
        VALUES(88,'Mello Jello','2 min','The What',4);
```

4. Create a view based on the DJs on Demand COPY_D_CDS table. Name the view read_copy_d_cds. Select all columns to be included in the view. Add a WHERE clause to restrict the year to 2000. Add the WITH READ ONLY option.

```
CREATE OR REPLACE VIEW read_copy_d_cds  AS
SELECT *
FROM copy_d_cds
WHERE year = '2000'
WITH READ ONLY ;

SELECT * FROM read_copy_d_cds;
```

5. Using the read_copy_d_cds view, execute a DELETE FROM red_copy_d_cds WHERE cd_number = 90;

```
ORA-42399: cannot perform a DML operation on a read-only view
```

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 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?

    7

11. What are the restrictions on modifying data through a view?

    Delete restricted if contains group functions, group by clause, distinct. rownum

12. What is Moore's law? Do you consider that it will continue to apply indefinitely? Support your opinion.

    Computing power nearly doubles every year

13. What is the singularity in terms of computing?
    Super computer will lead to overgrowth and become uncontrollable that will irreversibilly affect human civilization

Practice 15-3

Top - n analysis	Asks for the N largest or smallest values in a column
Drop view view_name	Removes a view
Inline view	Subquery with an alias that can be used within a SQL statement

1. Created 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 OR REPLACE 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;
ORA-00942: table or view does not exist
```

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 * FROM
(SELECT last_name, salary FROM employees ORDER BY salary  DESC)
WHERE ROWNUM <= 3;
```

4. Construct an inline view from the Oracle database that lists the last name, salary, department ID, and maximum salary for each department.

SELECT empm.last_name, empm.salary, dptmx.department_id

```
FROM
(SELECT dpt.department_id, MAX(NVL(emp.salary,0)) max_dpt_sal
FROM departments dpt LEFT OUTER JOIN employees emp ON dpt.department_id =
emp.department_id
GROUP BY dpt.department_id) dptmx LEFT OUTER JOIN employees empm ON
dptmx.department_id = empm.department_id
WHERE NVL(empm.salary,0) = dptmx.max_dpt_sal;
```

5. Create a query that will return the staff members of GLobal Fast Foods ranked by salary from lowest to highest.

```
SELECT ROWNUM,last_name, salary
FROM
(SELECT * FROM f_staffs ORDER BY SALARY);
```

Practice 16-1

Create sequence	Command that automatically generates sequential numbers
sequences	Generates a numeric value
nextval	Returns the next available sequence value
Increment by	Specifies the interval between sequence numbers
nomaxvalue	Specifies a maximum value fo 10^27 for an ascending sequence and -1 for a descending sequence (default)
currval	Returns the current sequence value
minvalue	Specifies the minimum sequence value
cycle/ no cycle	Specifies whether the sequence continues to generate values after reaching its maximum or minimum values
nominvalue	Specifies a minimum value fo -1 for an ascending sequence and -(10^26) for a descending sequence (default)
maxvalue/ nomaxvalue	Specifies a maximum or default the sequence can generate
Start with	Specifies the first sequence number to be generated
cache/ nocache	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.

```
CREATE TABLE seq_d_songs
AS ( SELECT * FROM d_songs);
DESCRIBE seq_d_songs;
DESCRIBE d_songs;
SELECT * FROM d_songs;
SELECT * FROM seq_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
INCREMENT BY 2
START WITH 100
MAXVALUE 1000
NOCYCLE
NOCACHE;
```

3. Query the USER_SEQUENCES data dictionary to verify the seq_d_songs_seq SEQUENCE settings.

```
SELECT * FROM user_sequences WHERE sequence_name = UPPER('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.

```
INSERT INTO seq_d_songs (id,title,duration,artist,type_code)
VALUES(seq_d_songs_seq.NEXTVAL,'Surfing Summer',NULL,NULL,12);
INSERT INTO seq_d_songs (id,title,duration,artist,type_code)
VALUES(seq_d_songs_seq.NEXTVAL,'Victory Victory','5 min',NULL,12);
SELECT * FROM seq_d_songs ORDER BY id DESC;
SELECT * FROM user_sequences WHERE sequence_name = UPPER('seq_d_songs_seq');
```

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?

   Generate identity column values

7. What are the advantages of caching sequence values?

   Performance advantage

8. Name three reasons why gaps may occur in a sequence?

   Rolling back statement, same sequence being used by multiple tables

Practice 16-2

Confirming index	Confirms the existence of indexes from the USER_INDEXES data dictionary view
index	Schema object that speeds up retrieval of rows
Create public synonym	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
Function-based 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. Schema objects which make retrieval of rows from table faster--meant to be efficient way
2. What is a ROWID, and how is it used?
   a. Indexes use ROWIDs, fastest way to access rows
3. When will an index be created automatically?
   a. Can be created unique/primary constraint in the table. So, it means that primary key/unique key.

4. Create a non unique index (foreign key) for the DJs on Demand column (cd_number) in the D_TRACK_LISTINGS table. Use the Oracle Application Developer SQL WOrkship Data Browser to confirm that the index was created.

```
CREATE INDEX d_tlg_cd_number_fk_i
 on d_track_listings (cd_number);
```

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 ucm.index_name, ucm.column_name, ucm.column_position, uix.uniqueness
FROM user_indexes uix INNER JOIN user_ind_columns ucm ON uix.index_name =
ucm.index_name
WHERE ucm.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 PUBLIC SYNONYM dj_tracks FOR d_track_listings;
```

8. Create a function-based index for the last_name column in the DJs on Demand D_PARTNERS table that makes it possible not to have capitalize the table name for searches. Write a SELECT statement that would use this index.

```
CREATE INDEX d_ptr_last_name_idx
ON d_partners(LOWER(last_name));

SELECT *
FROM d_partners
WHERE LOWER(last_name) = 'something';
```

9. Create a synonym for the D_TRACK_LISTIINGS table. Confirm that is has been created by querying the data dictionary.

```
CREATE SYNONYM dj_tracks FOR d_track_listings;
ORA-00955: name is already used by an existing object
CREATE SYNONYM dj_tracks2 FOR d_track_listings;
```

Synonym created.
SELECT * FROM user_synonyms WHERE table_NAME = UPPER('d_track_listings');

10. Drop the synonym that you created in question 9.

    DROP SYNONYM dj_tracks2;

Synonym dropped.
SELECT * FROM user_synonyms WHERE table_NAME = UPPER('d_track_listings');

Practice 17-1

1. What are system privileges concerned with? - what user can at DB level
2. What are object privileges concerned with? - concerned with right to perform a particular action on an object or to access another user's object
3. What is another name for object security? Data security
4. What commands are necessary to allow Scott access to the database with a password of tiger?
   a. CREATE USER scott IDENTIFIED BY tiger
   b. GRANT CREATE SESSION TO scott;
   c. GRANT CREATE SESSION, CREATE TABLE, CREATE sequence, CREATE VIEW TO scott;
5. What are the commands to allow Scott to SELECT from and UPDATE the d_clients table?
   a. GRANT SELECT, UPDATE ON hkumar.d_clients to scott;
      Opposite: REVOKE SELECT, UPDATE ON d_clients FROM scott;
      Verify: SELECT * FROM user_tab_privs_made
6. What is the command to allow everybody the ability to view the d_songs table?
   GRANT SELECT ON hkumar.d_songs to PUBLIC;
   Opposite: REVOKE SELECT ON hkumar.d_songs FROM PUBLIC;
   Verify: SELECT * FROM user_tab_privs_made;

7. Query the data dictionary to view the object privileges granted to you the user.
   SELECT * from user_tab_privs_recd;
   DESCRIBE user_tab_privs_recd

8. What privilege should a user be given to create tables?
   CREATE TABLE

9. If you create a table, how can you pass along privileges to other users just to view your table?
   GRANT SELECT ON hkumar.d_songs to scott1, scott2, scott3;
   Verify: SELECT * FROM user_tab_privs_made;

10. What syntax would you use to grant another user access to your copy_employees table?

GRANT ALL ON hkumar.copy_employees to   scott1, scott2, scott3;x`
Verify: SELECT * FROM user_tab_privs_made;

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;
    DESCRIBE user_col_privs_rec

Practice 17-2

1. What is a role? - group of related privileges that can be granted to users
2. What are the advantages of a role to a DBA? Makes it easier to grant/revoke/maintain 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 hkumar.d_clients to strange_uname
    WITH GRANT OPTION

    REVOKE SELECT ON hkumar.d_clients FROM strange_uname

    Verify: SELECT * FROM user_tab_privs_made;

SELECT * FROM user_tab_privs_made;

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?

    Group privileges in role and grant this role to user as appropriate

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?
   Defines a one-way communication path from one oracle DB to another oracle db

Practice 17-3

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.
      Describe locations;
      SELECT constraint_name, constraint_type, r_constraint_name FROM user_constraints
      WHERE table_name = 'LOCATIONS';
      SELECT constraint_name, constraint_type, table_name FROM user_constraints
      WHERE r_constraint_name = (SELECT constraint_name FROM user_constraints
      WHERE table_name = 'LOCATIONS' AND constraint_type = 'P')
   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 street_address, REGEXP_REPLACE(street_address, ' ','')
street_address_changed
FROM locations**