26/9/25

EXERCISE 13

Creating Views

1. What are three uses for a view from a DBA's perspective?

* Restrict Data Access

* Simplify Complex Queries

* Provibe Data Independence

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 NEW view.d Songs AS
SELECT id, title AS "Song title", artist
 FROM d songs.
 WHERE type-code = 'New_Age';
```

3. SELECT * FROM view_d_songs. What was returned?

type - code = 'New Age'

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

```
CREATE OR REPLACE VIEW view.d. Songs(song_ID, Song_Ti
   Performer, Song_Type)
 SELECT id, title artist, type-code
  FROM d Songs
   WHERE type-code = 'New Age";
```

Or use alias after the CREATE statement as shown.

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.d
SELECT Event.name AS 'Event name",
    Event date AS 'Event Date",
    theme desc AS 'Theme';
FROM d_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_Summary AS
SELECT department_id AS 'Dept ID',
       MAX(salary) AS "Max Salary",
       AVG (Salary) AS "Avg Salary";
FROM employees;
GROUP BY department_id;
```

DML Operations and Views

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. All table names in the data dictionary are stored in uppercase.

```
SELECT Table. name , column. name, ins ertable, uptable
                                               deletable
FROM user. updatable -columns.
WHERE table-name IN ("COPY. D .SONGS", "COPY . D.E ', 'COPY.
                 -IDS', 'COPY. D . (CUNO'S);
```

Use the same syntax but change table_name of the other tables.

2. Use the CREATE or REPLACE option to create a view of *all* the columns in the copy_d_songs table called view_copy_d_songs.

```
CREATE  OR REPLACE  VIEW view. copy d Songs AS
Select * FROM  copy . d. songs
```

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

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

```
INSERT INTO view-copy. d-songs (id, title, de duratio
artist type. code)
VALUES(88, 'Mello Jello', 2, 'The What', 4);
     SELECT * FROM Copy. d- Songs;
```

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 years = 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 - 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
SELECT * FROM copy.d code
WHERE year = 2000
WITH CHECK OPTION CONSTRAINT

7. Use the read_copy_d_cds view to delete any CD of year 2000 from the underlying copy_d_cds.

DELECTE RROM 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 = 200;

10. Execute a SELECT * statement for the base table copy_d_cds. What rows were deleted?

SELECT * FROM copy.d.cds

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

* contians GROUP BY DISTRICT or aggrgole f
* containe expression or joins
* Based of multiple tasks.

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

It states that the number of transistor on a microship every two years, heal and equation effects are showing it moores' law is approching its end in traditional silicon techology.

13. What is the "singularity" in terms of computing?

The techological singularity is a hypotentical future point where artitical intelligence suspeses human inteligene, leading uncontrollable technologial growth.

Managing Views

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 OR REPLACE VIEW _copy - d - Song - As
SELECT , dle , o dert
FROM copy - d _ Songe ;
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 . copy - d - songs;
SECECT 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.

```
SEIECT last - name - Salary - RANK()
Salary - rant
FROM employees
WHERE C = 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 l.last-name , e - Salary , e . department _id,
m. max -
    FROM department
    JOIN _ AS max - Salary FROM employee
```

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

```
SELECT name, salary
Salary ASC . AS rank - by Salory
FROOM global - post - foods . staff;
```

Indexes and Synonyms

1. What is an index and what is it used for?

An index is a database object that improves the Speed of data retrieval a table. it works lik a book index access without scanning th whole table.

2. What is a ROWID, and how is it used?

ROWID is a unique address of a row in a table its used to lacote rows quickly.

3. When will an index be created automatically?

An indexc is automatically created when a column is decaleared as Primary key or unique Constraint.

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 Express SQL Workshop Data Browser to confirm that the index was created.

```
Creat Index d_cd_idx
on - d - track - distings.
```

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 index_name, table_name, uniques
FROM user_indexes

WHERE 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 tables_names ='D_Events';
```

7. Write a query to create a synonym called dj_tracks for the DJs on Demand d_track_listings table.

```
Create SYNONYM dy_tracks FOR d_track_
                                    Litings;
```

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 partners_lname_idx;
ON d_partness (UPPER (last_name));
SELECT * FROM d_partners;
WHERE UPPER (last_nad ):
140
```

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

CREATE SYNONYM dj_tracks FOR d_track_listings
FROM user. Synonym WHERE synonyn.name = 'DJ_Tracks';

10.Drop the synonym that you created in question

DROP SYNONYM dj_tracks.

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	