2/9/25

EXERCISE 13

Creating Views

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

- Security
- Simplifies queries
- 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 view view_d_songs as
select id, title as "song title", artist
from DJ_on_demand
where type_code = "New age"
```

3. SELECT * FROM view_d_songs. What was returned?

returns id, song title for "new age" songs

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

```
Create view_d_song as
select id as "ID" title as "song title, artist as "Artist",
type code as "type code" from DJs_on_demand
where type_code = "New age";
```

Or use alias after the CREATE statement as shown.

```
Create view view_d_songs ("ID", "song title", "Artist", "type
code") as select id, title, artist, 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 view view_event_list as
select event_name as "Event name", event_date,
theme_desc as "Theme_description" 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 view view_dept_salary as
select dept_id as "department ID",
min(salary), max(salary), avg(salary)
from employees
group by dept_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 * from user_updatable_columns
where table_name = 'copy_d_songs';
```

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 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 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 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 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
```

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

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

Read only view can't be changed
Cannot update derived or Computed Columns

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

The no. of transistors on a chip doubles about every two years. it is slowing down due to physical limits

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

The technological singularity is a point where AI surpasses human intellegence, leading to rapid uncontrollable technological 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 view view_d_copy_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_d_copy_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,
from employees
order by salary desc
fetch first 3 rows only
```

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, d.max_salary
from employees e join (select dept_id, max(salary) as max_salary
from employees group by dept_id) d on e.dept_id = d.dept_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 from staff
order by salary ASC;
```

Indexes and Synonyms

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

it is a database object that improves the speed of data retrie in table

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

Rowid in a unique adress of a row in a table; it is used to locate raws

3. When will an index be created automatically?

when a primary key or unique Constraint is defined on a Colum

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.

```
Create index idx - cd_numbers
on d_track - listing (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 i.index_name, i.table_name, i.uniqueness
from user_indexes i
where i.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_index
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 - upper_lastname
on d_partners (upper(last_name));
select * from d_partners
where upper(last_name) = 'SMITH';
```

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

Create synonym track_list for d_track_listings

select synonym_name, table_name from user_synonyms

where synonym_name = 'Track List';

10. Drop the synonym that you created in question

Drop Synonym track_list;

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