EXERCISE 13

Creating Views

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

Security : Restrict access to sensitive data

simplification : Hide complex joins and logic

Data independence : Abstract schema changes from users.

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 d-songs where type_code = 'New Age';
Query the view;

3. SELECT * FROM view_d_songs. What was returned?

ID, Song title, Artist for songs with type_code = 'NewAge'.
replace view with type_code and aliases.

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

create or replace view_d_songs as select id As "song ID", title as "song title", artist as "Performer", type_code as "Genre" 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 view view_event_schedule as select name as "Event Name",
event_date AS "Date", description as "Theme"
From d_events; view for department managers.

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 dept_salary_summary as select department_id,
min (salary) as min_salary, 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 column _name , uptable from User_ Uptable _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 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 ,duration, artist,
type-code)
values (88 , 'Mello jello', 2, '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.

select * from copy_d_songs;

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

create view read_copy_a_cds as select * from copy_d_cds where year = 2000 with read only;

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.

Delete from read_copy_d_cds where cd_number = 90;

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 ;

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

view must be updatable
cannot modify data
READ ONLY views block DML .

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

No. of transistors on a chip doubles every 2 years .

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

Hypothetical future point where AI surpasses human intelligence .

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.

```
Select title , artist from copy-d-songs ;
```

2. Issue a DROP view_copy_d_songs. Execute a SELECT * statement to verify that the view has been deleted.

```
select * from view_copy-d-songs ;
Drop the view
```

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
```

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.deparment.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 first_name , last_name , salary FROM f_staffs ORDER BY
salary ASC ;
```

✓

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 operations on a table. It works like a lookup table, allowing Oracle to find rows faster without scanning the entire table.

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

ROWID is a unique identifier for each row in a table, representing its physical location. It's used internally by Oracl for fast access and can be selected to identify or update specific rows.

3. When will an index be created automatically?

* A PRIMARY KEY or UNIQUE constraint is defined. * A foreign key is created. * Certain function-based or bitmap indexes 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 Express SQL Workshop Data Browser to confirm that the index was created.

```
CREATE INDEX idx_cd_number 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 i.index_name, i.table_name, i.uniqueness FROM
user_indexes. i JOIN user_ind_columns c ON i.index_name
= c.index_name 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_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_lower_lastname ON d_partners (LOWER
      (last_name));

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

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

```
CREATE SYNONYM track_syn FOR d_track_listings;
SELECT synonym_name, table_name FROM user_synonyms
WHERE synonym_name = 'TRACK_SYN';
```

10.Drop the synonym that you created in question

```
DROP SYNONYM track_syn;
```

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