

EXERCISE 13

Creating Views

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

1. security - restricts access of specific column or row of data.
2. simplification - simplify complex SQL queries by storing them as a single view.

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"  
FROM dis-on-demand WHERE type-code = 'New age';
```

3. SELECT * FROM view_d_songs. What was returned?

This query will return all rows from the view_d-song view, showing only . id
• artist.

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

```
CREATE OR REPLACE VIEW view_dj-song AS  
SELECT id AS "Song ID"  
       title AS "song title"  
       artist AS "Artist name"  
FROM djs-on-demand  
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_list AS SELECT
event_name AS "event name"
event_date AS "Event date"
theme_description AS "Theme description"
```

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_salaries AS
SELECT department_id AS "DEPARTMENT-ID",
MIN SALARY (salary) AS "MIN SALARY",
MAX SALARY (salary) AS "MAX SALARY",
AVG SALARY (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,  
FROM user update table-column  
WHERE table-name = 'COPY-D-SONGS',  
COPY-D-EVENTS, COPY-D-CDS, COPY-D-CLIENTS;
```

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', M);  
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 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;

```
SELECT 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 OR REPLACE VIEW read-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 = 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?

- * Contains group by distinct or aggregate functions
- * Contains expressions or joins
- * Contains with Read only
- * Based on multiple tables
- * Violates with check option

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 no. of transistors on a microchip doubles approximately every two years while the cost of computers is halved. No. physical limits of silicon, heat and quantum, effects are slowing it.

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

The technological singularity is a hypothetical future point where AI surpasses human intelligence, leading to rapid 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 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;
```

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, rank() over (order by salary desc)  
as salary_rank from employees where column <= 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 e.last_name, e.salary, e.department, m.max_salary  
from employees e join (select department_id, max_salary from  
employees);
```

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

```
select name, salary, rank() over (order by salary asc)  
as rank-by-salary from global_fast_foods_staff;  
group by department_id) m  
on e.department_id = m.department_id;
```


Indexes and Synonyms

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

An index is a database object that improves query performance by allowing faster retrieval of rows using pointers to physical data.

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

ROWID is a unique assigned to each row in an Oracle table. It identifies the exact physical storage of row in db.

3. When will an index be created automatically?

primary key or unique constraint is defined on a column

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)
```

SELECT index-name, table_name FROM user_indexes

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, uniqueness  
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 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-last-name  
on d-partners (upper-last-name);  
select * from d-partners;
```

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

Create synonym tracks for d-tracks-listings;
select synonyms_name, table_owner, table_name
FROM user_synonyms where synonyms_name = 'TRACKS';
10. Drop the synonym that you created in question

OR synonym-tracks;

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