

Date: 2/9/25

### EXERCISE 13

#### Creating Views

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

1. Security - Restrict access to specific columns or rows of data.
2. Simplification - Simplify complex SQL queries by storing them on single view.
3. Data Consistency - Provide a consistent, unchanging interface even if underlying.

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-a-songs AS SELECT  
id, title AS 'Song Title', artist AS djs-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-a-songs view, showing only \* id \* song title \* artist.

4. REPLACE view\_d\_songs. Add type\_code to the column list. Use aliases for all columns.

```
CREATE OR REPLACE VIEW view-a-songs AS  
SELECT id AS 'Song ID', title AS 'Song  
Title', artist as 'Artist Name', type-code  
AS 'Type-code', 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 views-event-list AS  
SELECT event-name AS "Event Name",  
event-date AS "Event Date", theme-description  
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-salaries AS  
SELECT department-id AS "Department  
ID", MIN(salary) AS "Minimum Salary",  
MAX(salary) AS "Maximum Salary",  
AVG(salary) AS "Average 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, insertable,  
updatable, deletable FROM user-updatable-  
columns 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-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 CDs 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;

```
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 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  
cd-number = 90;
```

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;
```

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;  
see rows from copy\_d\_cds.

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

1. The view must be updatable. 2. If the view has with READ ONLY, no insert, UPDATE, OR DELETE is allowed.

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

Moore's Law: Moore's Law states that the number of transistors on a microchip doubles approximately every 18 to 20 months, which results in computers becoming faster and adapting over time.

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

the technological singularity is a theoretical point in the future when artificial intelligence surpasses human intelligence to the extent that it can improve itself without human input at this point.

### 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_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 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. Hint: One query will need to calculate maximum salary by department ID.

```
SELECT e.last_name, e.salary, e.department_id,  
m.max_salary FROM employees e JOIN (SELECT  
department_id, MAX(salary) AS 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;
```

## 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 locations.

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

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

3. When will an index be created automatically?

When a primary key or unique constraint of -frack-listings!

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 of-frack-listings  
(cd_number); SELECT index_name, table_name FROM  
user_indexes WHERE table_name = 'D-TRACK-LISTINGS';
```

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));
```

141

```
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 tracks FOR d-track-listings;  
SELECT synonym_name, table_owner, table_name  
FROM user-synonyms WHERE synonym_name = 'TRACKS';
```

10. Drop the synonym that you created in question

```
DROP SYNONYM tracks;
```

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