

EXERCISE 13 Creating Views

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

1. security - use strict access of specific column or rows of data

2. simplification - simplify complex SQL queries by storing them as a single view.

3. data consistency - Provides a consistent, unchanging image even if underlying table or structure changes.

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 dis-on-demand WHERE type_code = 'New Age';
```

3. SELECT * FROM view_d_songs. What was returned?

This query will return all from the view-d-song view, showing only:
• id
• song title
• artist
only for those where the data = "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-song AS
SELECT id AS "Song-ID",
       title AS "Song Title",
       artist AS "Artist Name",
       type_code AS "Type Code",
FROM dis-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 - is 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 VIEWS 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
```

DML Operations and Views

GROUP BY department_id;

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, updateable, deletable
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
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);
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
AS select *
FROM copy - d - cds
WITH CHECK OPTION CONSTRAINT
ck - read - copy - d - cds;
select * from 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.

```
SELECT FROM read_copy_d_cds;
```

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;
```

All rows from copy_d_cds where year > 2000 were deleted

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

you can't change or delete data through a view
if the view is based on multiple tables

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

un power about every two years. It says computer doubles but is slowing down now.

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

The "singularity" is when computers become smarter than humans. At that point. This called cause very fast changes in Technology.

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_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  
FROM employee  
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 employee  
join
```

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

```
select * from staff_rank_order
```

Indexes and Synonyms

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

A tool that speeds up searching for data in table.
It is used for finding data.

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

The physical address of a row in the disk.

3. When will an index be created automatically?

When you set a column as primary key.

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 d-track-listings-cd-ix

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.uniqueness
C.COLUMN-NAME

WHERE
i.table-name = 'D_SONGS'

ORDER BY
1. index-name, column

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

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 d_partners_idx on d_partners (upper(last_name));

select first_name, last_name, partners_id
from d_partners
where upper(last_name)
```

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

```
create synonym dt_tracks
for D_TRACK_LISTINGS
```

10. Drop the synonym that you created in question

```
select synonym_name,
table_owner,
table_name
```

```
from user_synonyms
where synonym_name = 'dt_tracks'
```