

Database Programming with SQL

15-1: Creating Views

1. What are three uses for a view from a DBA's perspective?
 - Data Security and Access Control
 - Simplifying Complex Queries
 - Data Abstraction and Consistency
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.

```
1 CREATE OR REPLACE VIEW view_d_songs AS
2 SELECT id AS song_id, title AS "Song Title", artist
3 FROM d_songs
4 WHERE type_code = 'New Age';
```

Results Explain Describe Saved SQL History

View created.

0.02 seconds

3. SELECT * FROM view_d_songs. What was returned?

```
1 SELECT * FROM view_d_songs
2
```

Results Explain Describe Saved SQL History

SONG_ID	Song Title	ARTIST
45	Its Finally Over	The Hobbits
46	Im Going to Miss My Teacher	Jane Pop

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

```
1 CREATE OR REPLACE VIEW view_d_songs AS
2 SELECT
3     id AS song_id,
4     title AS "Song Title",
5     artist AS song_artist,
6     type_code AS song_type
7 FROM d_songs
8 WHERE type_code = 'New Age';
```

Results Explain Describe Saved SQL History

View created.

- ```
1 select * from event_setup_view
```

Results Explain Describe Saved SQL History

| Event Name        | Event Date  | Theme                                 |
|-------------------|-------------|---------------------------------------|
| Peters Graduation | 14-May-2004 | Party for 200, red, white, blue motif |
| Vigil wedding     | 28-Apr-2004 | Black tie at Four Season hotel        |

- [illegible]

1. Query the data dictionary `USER_UPDATABLE_COLUMNS` to make sure the columns in the base tables will allow `UPDATE`, `INSERT`, or `DELETE`. Use a `SELECT` statement. All table names in the data dictionary are stored in uppercase.

```
1 SELECT table_name, column_name, insertable, updatable, deletable
2 FROM user_updatable_columns
3 WHERE table_name = 'COPY_D_SONGS';
```

Results

Explain

Describe

Saved SQL

History

| TABLE_NAME   | COLUMN_NAME | INSERTABLE | UPDATABLE | DELETABLE |
|--------------|-------------|------------|-----------|-----------|
| COPY_D_SONGS | ID          | YES        | YES       | YES       |
| COPY_D_SONGS | TITLE       | YES        | YES       | YES       |
| COPY_D_SONGS | DURATION    | YES        | YES       | YES       |
| COPY_D_SONGS | TYPE_CODE   | YES        | YES       | YES       |

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.

```
1 CREATE OR REPLACE VIEW view_copy_d_songs AS
2 SELECT *
3 FROM copy_d_songs;
```

**Results** Explain Describe Saved SQL History

View created.

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         |

```
1 INSERT INTO view_copy_d_songs (id, title, duration, artist, type_code)
2 VALUES (88, 'Mello Jello', '2', 'The What', 4);
```

**Results** Explain Describe Saved SQL History

1 row(s) inserted.

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.

```
1 CREATE OR REPLACE VIEW read_copy_d_cds AS
2 SELECT *
3 FROM copy_d_cds
4 WHERE year = 2000
5 WITH READ ONLY;
```

**Results** Explain Describe Saved SQL History

View created.

5. Using the read\_copy\_d\_cds view, execute a DELETE FROM read\_copy\_d\_cds WHERE cd\_number = 90;
  - Error at line 1/13: ORA-42399: cannot perform a DML operation on a read-only view
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.
  - Table View was created but no results were found since there are no records from year 2000.
7. Use the read\_copy\_d\_cds view to delete any CD of year 2000 from the underlying copy\_d\_cds.

```
1 DELETE FROM read_copy_d_cds;
2
```

| Results           | Explain | Describe | Saved SQL | History |
|-------------------|---------|----------|-----------|---------|
| 0 row(s) deleted. |         |          |           |         |

8. Use the read\_copy\_d\_cds view to delete cd\_number 90 from the underlying copy\_d\_cds table.

```
1 DELETE FROM read_copy_d_cds
2 WHERE cd_number = 90;
```

| Results           | Explain | Describe | Saved SQL |
|-------------------|---------|----------|-----------|
| 0 row(s) deleted. |         |          |           |

9. Use the read\_copy\_d\_cds view to delete year 2001 records.

```
1 DELETE FROM read_copy_d_cds
2 WHERE year = 2001;
```

| Results           | Explain | Describe | Saved SQL | His |
|-------------------|---------|----------|-----------|-----|
| 0 row(s) deleted. |         |          |           |     |

10. Execute a SELECT \* statement for the base table copy\_d\_cds. What rows were deleted?

```
1 SELECT *
2 FROM copy_d_cds;
```

| CD_NUMBER | TITLE                      | PRODUCER         | YEAR |
|-----------|----------------------------|------------------|------|
| 98        | Holiday Tunes for All Ages | Tunes are Us     | 2003 |
| 97        | Celebrate the Day          | R & B Inc.       | 2003 |
| 99        | Party Music                | Old Town Records | 2004 |
| 100       | Best of Rock and Roll      | Old Town Records | 2004 |

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

- Read-only Views: No DML operations are allowed.
- With Check Option: Ensure that modifications via the view adhere to the view's WHERE clause conditions.

12. What is Moore's Law? Do you consider that it will continue to apply indefinitely?

- According to Gordon Moore's Law, computer power will increase exponentially as the number of transistors on a chip doubles roughly every two years.

Moore's Law is challenged by physical and quantum constraints, such as transistor downsizing approaching atomic scale.

Though not in the conventional sense, performance gains may be maintained by recent advancements in parallel computing, AI accelerators and alternative materials.

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

- The term "Singularity" describes a speculative future in which advances in technology, especially artificial intelligence (AI), become unstoppable and irreversible, causing significant disruptions to civilization. This idea frequently corresponds with AI.

## 15-3: 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.

```
1 SELECT * FROM view_copy_d_songs;
```

| TITLE           | ARTIST   |
|-----------------|----------|
| Mello Jello     | The What |
| Victory Victory | -        |
| Surfing Summer  | -        |

- Issue a DROP view\_copy\_d\_songs. Execute a SELECT \* statement to verify that the view has been deleted.

```
1 DROP VIEW view_copy_d_songs;
```

Results Explain Describe Saved SQL

View dropped.

- 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.

```
1 SELECT last_name, salary
2 FROM (
3 SELECT last_name, salary,
4 RANK() OVER (ORDER BY salary DESC) AS salary_rank
5 FROM employees)
6 WHERE salary_rank <= 3;
```

Results Explain Describe Saved SQL History

| LAST_NAME | SALARY |
|-----------|--------|
| King      | 24000  |
| Kochhar   | 17000  |
| De Haan   | 17000  |


- 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.


```
1 SELECT last_name, salary, department_id, max_salary
2 FROM (
3 SELECT last_name, salary, department_id,
4 MAX(salary) OVER (PARTITION BY department_id) AS max_salary
5 FROM employees)
6 ORDER BY department_id;
```


Results Explain Describe Saved SQL History


| LAST_NAME | SALARY | DEPARTMENT_ID | MAX_SALARY |
|-----------|--------|---------------|------------|
| Whalen    | 4400   | 10            | 4400       |
| Hernandez | 4300   | 10            | 4400       |
| Ricci     | 4100   | 10            | 4400       |
| Saikawa   | 4400   | 10            | 4400       |
| Stocks    | 3700   | 20            | 13000      |

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











A::





1 SELECT last\_name, salary

2 FROM f\_staffs

3 ORDER BY salary ASC;

Results

Explain

Describe

Saved SQL

History

| LAST_NAME | SALARY |
|-----------|--------|
| Doe       | 6.75   |
| Miller    | 10     |
| Tuttle    | 60     |