

Date: 24/9/25

EXERCISE 13

Creating Views

AIM: TO Create views in mySQL.

1. What are three uses for a view from a DBA's perspective?
 1. security and Access Control (Data Hiding).
 2. simplification / clarity
 3. logical data Independence

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';

3. SELECT * FROM view_d_songs. What was returned?

The database returned all the columns and rows from the d_songs table where the type_code equals '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_songs (song_id,
song_title, song_artist, song_type) AS select id, title,
artist, type_code FROM d_songs where type_code = 'New Age';

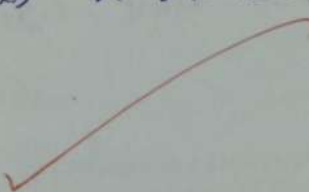
Or use alias after the CREATE statement as shown.

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-jason-events (event_name, event_date,
theme_desc) AS Select e.name, e.event_date, t.theme_description
FROM d-events e JOIN d-themes t ON e.theme_code =
t.theme_code;
```

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-salary-summary (dept_name,
min_sal, max_sal, avg_sal) AS select d.name, MIN(e.salary),
MAX(e.salary), AVG(e.salary) FROM d-employees e JOIN
d-departments d ON e.department_id = d.idc GROUP
BY d.name;
```



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, updatable, insertable FROM  
user_updatable_columns where table_name IN ('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 id, title, duration, artist, type_code 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);
```

```
COMMIT;
```

```
Select * FROM copy-d-songs where id = 88;
```


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 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;  
COMMIT;
```

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;  
COMMIT;
```

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; The songs with year = 2000 were deleted.

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

The view is generally ^{not} updatable if the view's defining query contains Aggregate Functions, GROUP BY CLAUSE, DISTINCT keyword, Complex Joins, Set Operations and ^{delete} insert only.

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 is the observation that the number of transistors in a dense integrated circuit (IC) doubles approximately every two years. Most experts agree that Moore's law, in its traditional sense, is reaching its physical limits.

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

Singularity in terms of computing: the term specifically refers to the point where 1. An AGI or super intelligence is created. 2. The new intelligence can autonomously improve itself at an exponential rate.

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 l_name, salary, RANK() OVER (ORDER BY salary DESC)
as salary_rank FROM employees where RANK() over (order BY salary
DESC) <= 3

Select l_name, salary, salary_rank FROM (Select l_name,
salary, RANK() OVER (ORDER BY salary DESC) as salary_rank
FROM employees) where salary_rank <= 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.l_name, e.sal, e.dept-id, d-max.max-dept-salary
FROM employees e JOIN (select dept-id, MAX(salary) AS
max-dept-salary FROM employees GROUP BY dept-id) d-max
ON e.dept-id = d-max.dept-id;

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

Select staff-member-name, salary, RANK() OVER
(ORDER BY salary ASC) AS rank-low-to-high FROM
global-fast-foods-staff order BY rank-low-to-high;

Indexes and Synonyms

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

An index is an optional schema object that is physically and logically independent of data in the table. It is used for speeding up data retrieval and enforcing uniqueness.

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

A ROWID is a pseudo-column column that represents the physical address of a row in the database. It is used by the database to locate rows rapidly and identify unique rows.

3. When will an index be created automatically?

An index is created automatically by the database when a primary key or a unique key constraint is defined on a column or set of columns in a table.

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-fk-idx 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.col_name, c.col_position
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 partners-lastname-lower-idx ON D-PARTNERS (LOWER(last_name));

```
select l-name, partner-id FROM D-PARTNERS where
LOWER(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 tracklist for D-TRACK-LISTINGS;
select synonym-name, table-owner, table-name FROM
USER-SYNONYMS where synonym-name = 'TRACKLIST';

10. Drop the synonym that you created in question

DROP SYNONYM tracklist;

RESULT:

Thus the views are created in MySQL

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	3
Total (15)	13
Faculty Signature	