

EXERCISE-13

Creating Views

Name: Vedhasree S

Register Number: 240701580

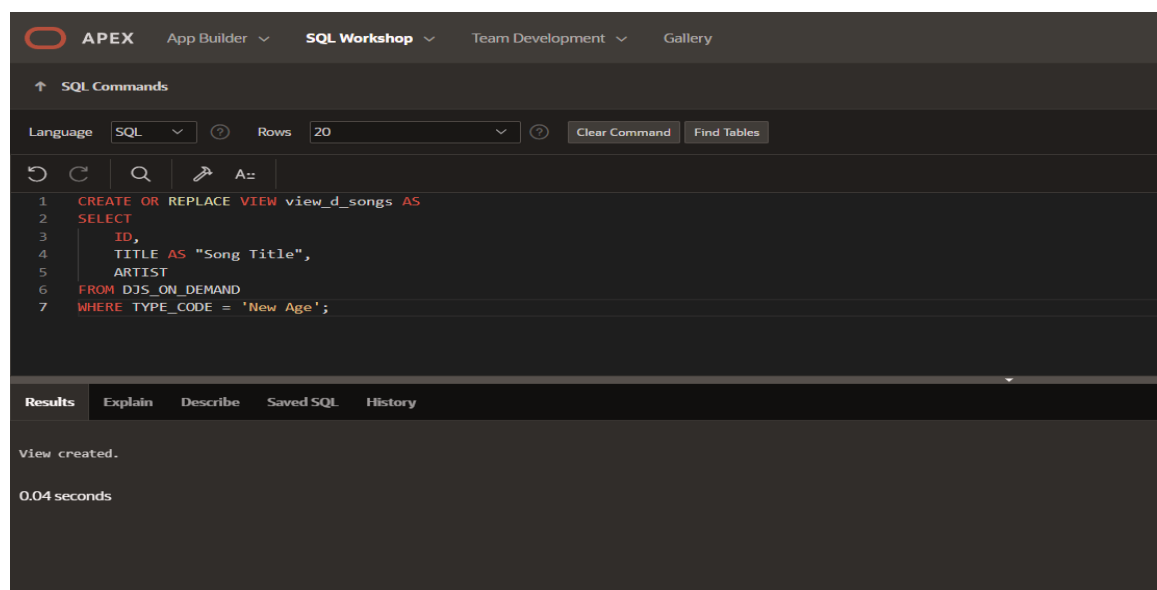
Department: CSE

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

Answer:

Three uses of a view from a DBA's perspective:

- Simplify complex queries: Views can hide complicated joins, filters, or calculations from the user. Instead of writing a long query every time, users can access a view that already presents the required data in a simple format.
 - Enhance security: Views can restrict access to sensitive information. For example, a view can show employees' names and departments without revealing their salaries, allowing managers or other users to see only what they are permitted to access.
 - Maintain data consistency and abstraction: Views provide a stable interface to the data. Even if the underlying table structures change, the view can be updated so that users' queries continue to work without modification.
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.

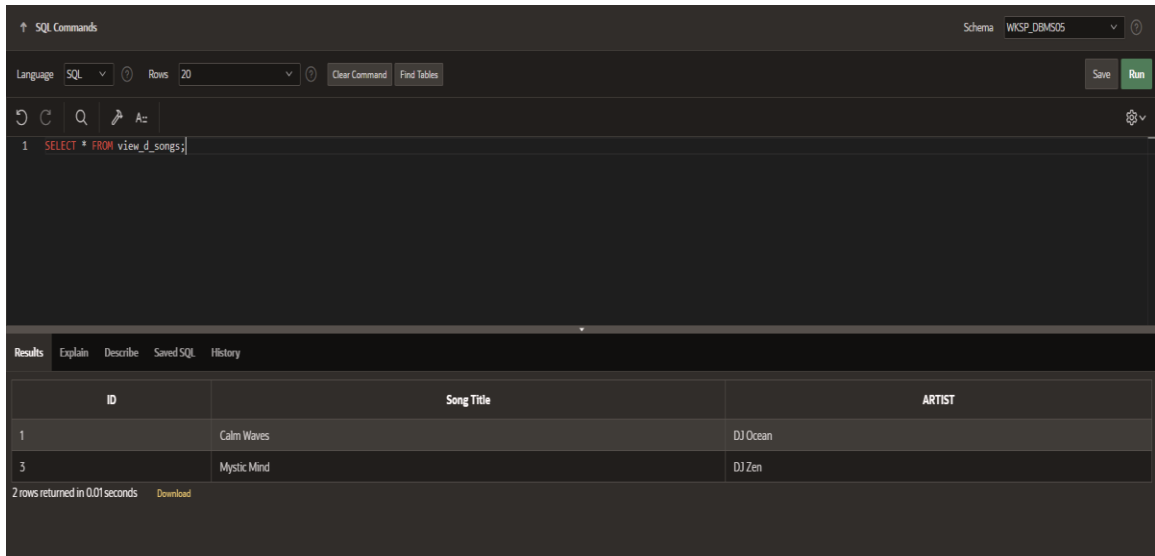


The screenshot shows the Oracle APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' section is active, displaying a SQL query to create a view named 'view_d_songs'. The query is as follows:

```
1 CREATE OR REPLACE VIEW view_d_songs AS
2 SELECT
3     ID,
4     TITLE AS "Song Title",
5     ARTIST
6 FROM DJs_ON_DEMAND
7 WHERE TYPE_CODE = 'New Age';
```

Below the query editor, the 'Results' tab is selected, showing the message 'View created.' and the execution time '0.04 seconds'.

3. `SELECT * FROM view_d_songs`. What was returned?

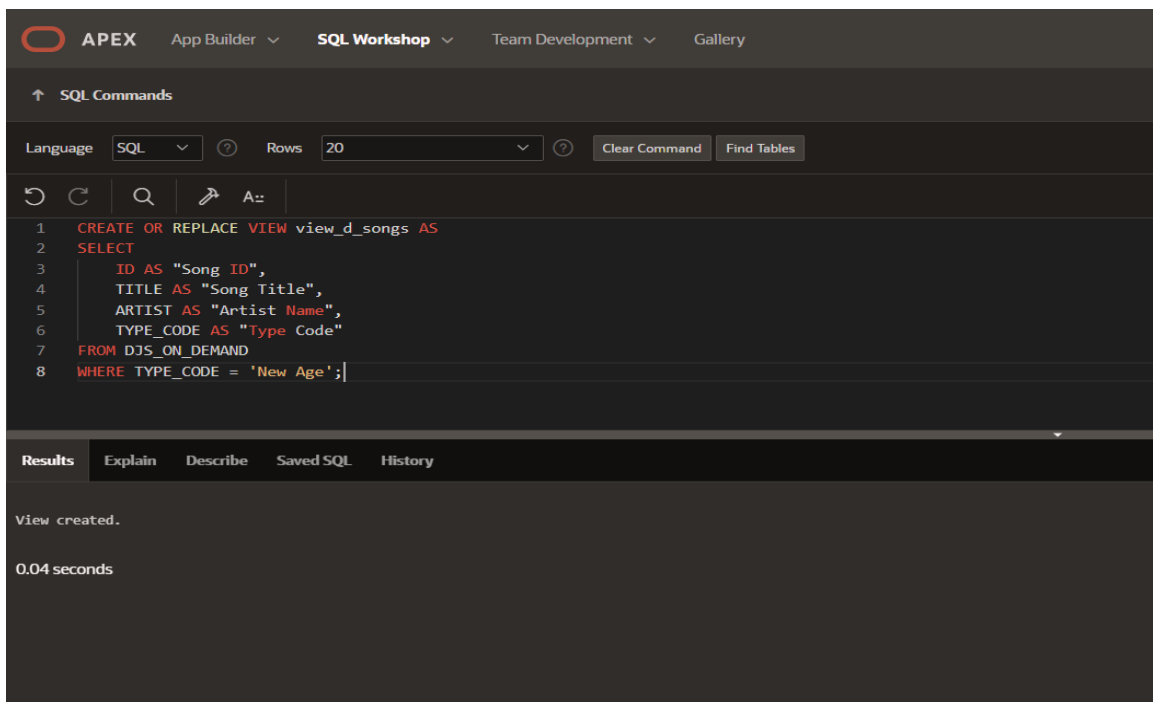


The screenshot shows the SQL Workshop interface with the command `SELECT * FROM view_d_songs;` entered. The results are displayed in a table with three columns: ID, Song Title, and ARTIST. Two rows are returned.

ID	Song Title	ARTIST
1	Calm Waves	DJ Ocean
3	Mystic Mind	DJ Zen

2 rows returned in 0.01 seconds

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



The screenshot shows the SQL Workshop interface with the command `CREATE OR REPLACE VIEW view_d_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';` entered. The results show "View created." and "0.04 seconds".

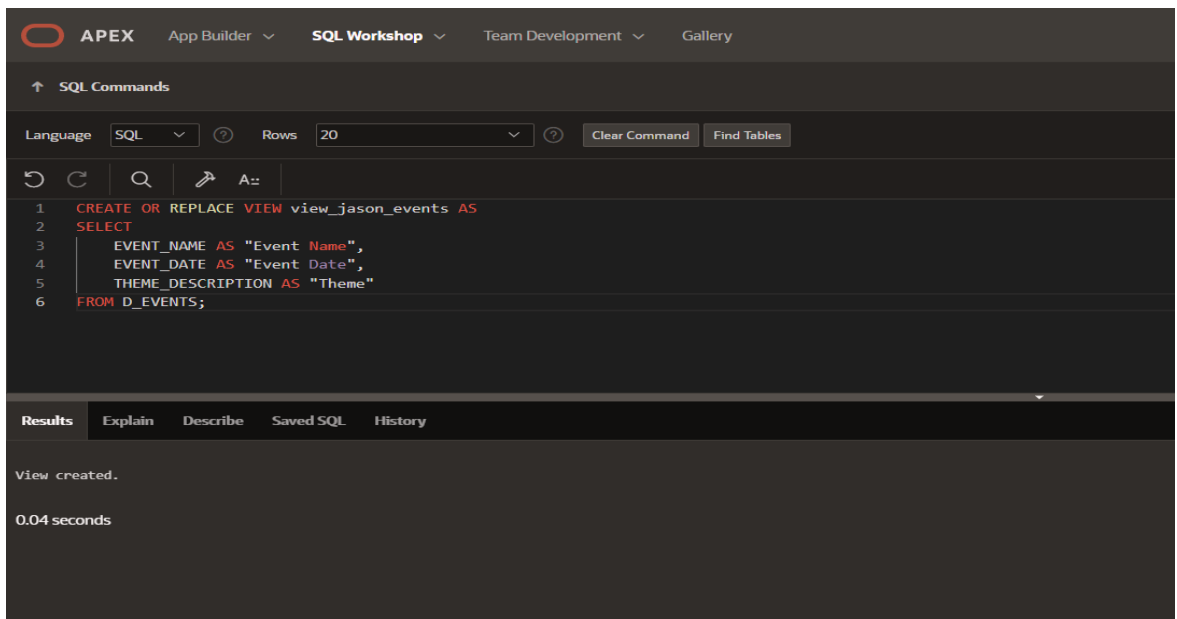
```
1 CREATE OR REPLACE VIEW view_d_songs AS
2 SELECT
3     ID AS "Song ID",
4     TITLE AS "Song Title",
5     ARTIST AS "Artist Name",
6     TYPE_CODE AS "Type Code"
7 FROM DJS_ON_DEMAND
8 WHERE TYPE_CODE = 'New Age';
```

View created.

0.04 seconds

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.

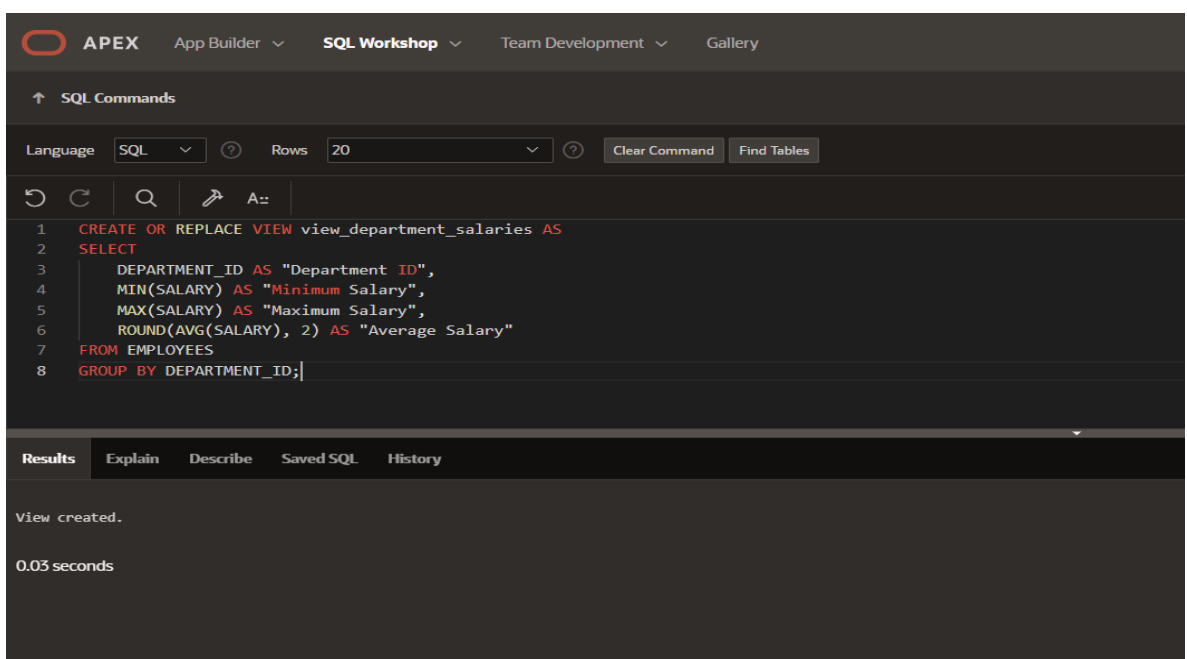


The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' section is active. The 'Language' dropdown is set to 'SQL', and the 'Rows' limit is set to '20'. The SQL command editor contains the following code:

```
1 CREATE OR REPLACE VIEW view_jason_events AS
2 SELECT
3     EVENT_NAME AS "Event Name",
4     EVENT_DATE AS "Event Date",
5     THEME_DESCRIPTION AS "Theme"
6 FROM D_EVENTS;
```

The 'Results' tab is selected, showing the message 'View created.' and the execution time '0.04 seconds'.

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.



The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' section is active. The 'Language' dropdown is set to 'SQL', and the 'Rows' limit is set to '20'. The SQL command editor contains the following code:

```
1 CREATE OR REPLACE VIEW view_department_salaries AS
2 SELECT
3     DEPARTMENT_ID AS "Department ID",
4     MIN(SALARY) AS "Minimum Salary",
5     MAX(SALARY) AS "Maximum Salary",
6     ROUND(AVG(SALARY), 2) AS "Average Salary"
7 FROM EMPLOYEES
8 GROUP BY DEPARTMENT_ID;
```

The 'Results' tab is selected, showing the message 'View created.' and the execution time '0.03 seconds'.

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.

The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying the command `DESC COPY_D_SONGS;`. The Results tab is selected, showing the table structure for `COPY_D_SONGS`.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COPY_D_SONGS	ID	NUMBER	22	-	-	1	-	-	-
	TITLE	VARCHAR2	100	-	-	-	✓	-	-
	DURATION	NUMBER	22	-	-	-	✓	-	-
	ARTIST	VARCHAR2	100	-	-	-	✓	-	-
	TYPE_CODE	VARCHAR2	20	-	-	-	✓	-	-

The screenshot shows the APEX SQL Workshop interface. The SQL Commands tab is active, displaying the command `DESC COPY_D_EVENTS;`. The Results tab is selected, showing the table structure for `COPY_D_EVENTS`.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COPY_D_EVENTS	EVENT_ID	NUMBER	22	-	-	1	-	-	-
	EVENT_NAME	VARCHAR2	100	-	-	-	✓	-	-
	EVENT_DATE	DATE	7	-	-	-	✓	-	-
	THEME_DESCRIPTION	VARCHAR2	100	-	-	-	✓	-	-
	PRICE	NUMBER	22	-	-	-	✓	-	-

APEX

App Builder

SQL Workshop

Team Development

Gallery

↑ SQL Commands

Language SQL Rows 20 Clear Command Find Tables

↶ ↷ 🔍 ↵ A::

1 DESC COPY_D_CDS;

Results Explain Describe Saved SQL History

Object Type TABLE Object COPY_D_CDS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COPY_D_CDS	CD_NUMBER	NUMBER	22	-	-	1	-	-	-
	CD_TITLE	VARCHAR2	100	-	-	-	✓	-	-
	YEAR	NUMBER	22	-	-	-	✓	-	-
	ARTIST	VARCHAR2	100	-	-	-	✓	-	-

APEX

App Builder

SQL Workshop

Team Development

Gallery

↑ SQL Commands

Language SQL Rows 20 Clear Command Find Tables

↶ ↷ 🔍 ↵ A::

1 DESC COPY_D_CLIENTS;

Results Explain Describe Saved SQL History

Object Type TABLE Object COPY_D_CLIENTS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COPY_D_CLIENTS	CLIENT_ID	NUMBER	22	-	-	1	-	-	-
	CLIENT_NAME	VARCHAR2	100	-	-	-	✓	-	-

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. Use the same syntax but change table_name of the other tables.

The first screenshot shows a SQL query in a dark-themed IDE. The query is: `SELECT TABLE_NAME, COLUMN_NAME, UPDATABLE, INSERTABLE, DELETABLE FROM USER_UPDATABLE_COLUMNS WHERE TABLE_NAME = 'COPY_D_SONGS';` The results are displayed in a table with 5 rows.

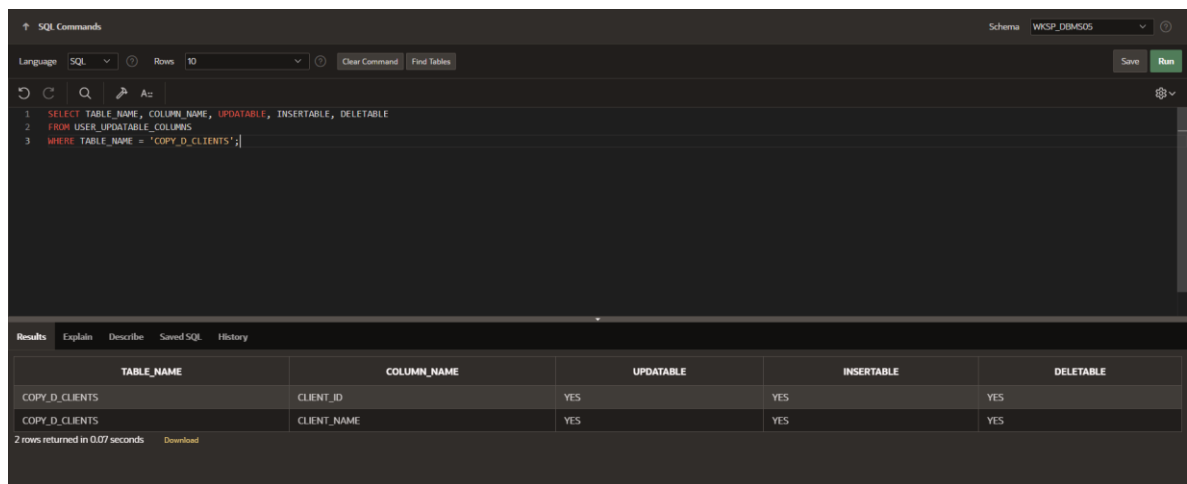
TABLE_NAME	COLUMN_NAME	UPDATABLE	INSERTABLE	DELETABLE
COPY_D_SONGS	ID	YES	YES	YES
COPY_D_SONGS	TITLE	YES	YES	YES
COPY_D_SONGS	DURATION	YES	YES	YES
COPY_D_SONGS	ARTIST	YES	YES	YES
COPY_D_SONGS	TYPE_CODE	YES	YES	YES

The second screenshot shows a similar query for the table 'COPY_D_EVENTS'. The results are displayed in a table with 5 rows.

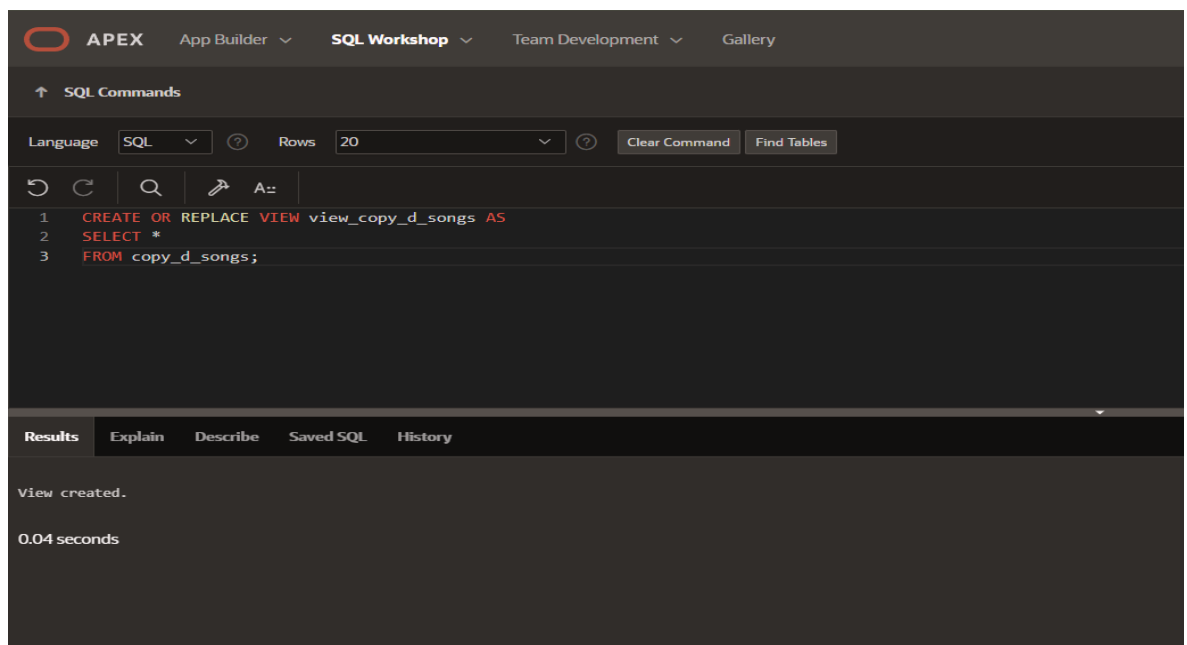
TABLE_NAME	COLUMN_NAME	UPDATABLE	INSERTABLE	DELETABLE
COPY_D_EVENTS	EVENT_ID	YES	YES	YES
COPY_D_EVENTS	EVENT_NAME	YES	YES	YES
COPY_D_EVENTS	EVENT_DATE	YES	YES	YES
COPY_D_EVENTS	THEME_DESCRIPTION	YES	YES	YES
COPY_D_EVENTS	PRICE	YES	YES	YES

The third screenshot shows a similar query for the table 'COPY_D_CDS'. The results are displayed in a table with 4 rows.

TABLE_NAME	COLUMN_NAME	UPDATABLE	INSERTABLE	DELETABLE
COPY_D_CDS	CD_NUMBER	YES	YES	YES
COPY_D_CDS	CD_TITLE	YES	YES	YES
COPY_D_CDS	YEAR	YES	YES	YES
COPY_D_CDS	ARTIST	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.



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

APEX

App Builder

SQL Workshop

Team Development

Gallery

↑ SQL Commands

Language SQL Rows 20 Clear Command Find Tables

Az

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.

0.03 seconds

APEX

App Builder

SQL Workshop

Team Development

Gallery

Q Search

VS

Vedhasree S

dbms05*

↑ SQL Commands

Schema WKSP_DBMS05

Language SQL Rows 20 Clear Command Find Tables Save Run

Az

1

SELECT * FROM copy_d_songs;

Results

Explain

Describe

Saved SQL

History

ID	TITLE	DURATION	ARTIST	TYPE_CODE
10	Dreamscape	3	DJ Dream	New Age
11	Happy Vibes	2	DJ Joy	Pop
88	Mello Jello	2	The What	4

3 rows returned in 0.01 seconds Download

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.

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The 'SQL Commands' section is active, displaying a SQL command to create a view. The command is: `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;`. Below the command, the 'Results' tab is selected, showing the message 'View created.' and the execution time '0.05 seconds'.

```
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.

0.05 seconds

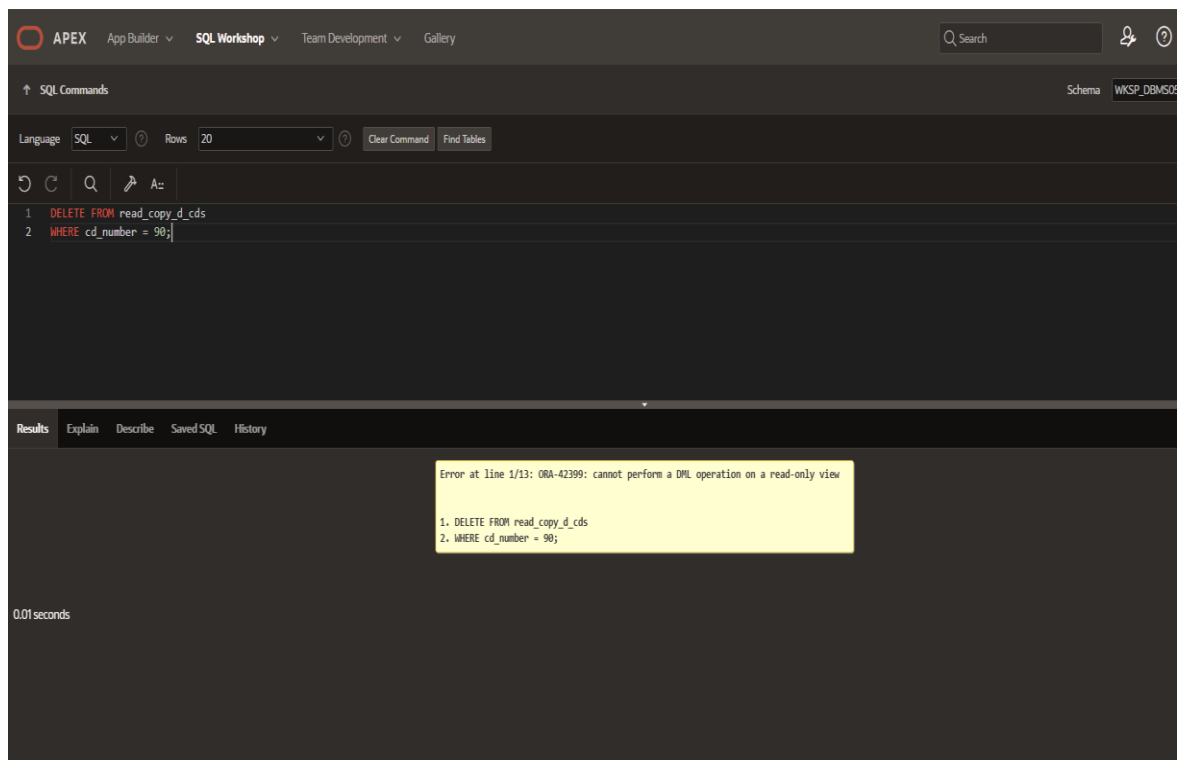
The screenshot shows the APEX SQL Workshop interface with the 'Results' tab selected. The SQL command entered is `1 SELECT * FROM read_copy_d_cds;`. The results are displayed in a table with four columns: CD_NUMBER, CD_TITLE, YEAR, and ARTIST. The table contains two rows of data. Below the table, it indicates '2 rows returned in 0.03 seconds' and provides a 'Download' link.

```
1 SELECT * FROM read_copy_d_cds;
```

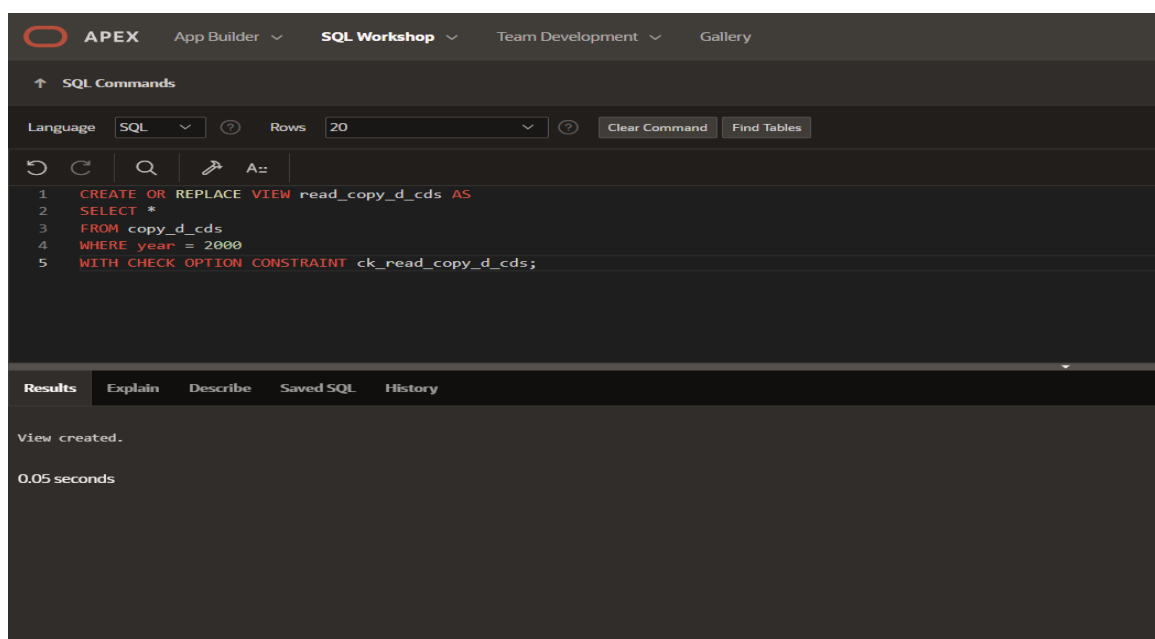
CD_NUMBER	CD_TITLE	YEAR	ARTIST
89	Soft Beats	2000	DJ Soft
90	Hard Rock	2000	DJ Hard

2 rows returned in 0.03 seconds [Download](#)

5. Using the `read_copy_d_cds` view, execute a `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.



SQL Commands

Schema: WKSP_DEMO5

Language: SQL Rows: 20 Clear Command Find Tables Save Run

```
1 SELECT * FROM read_copy_d_cds;
```

Results Explain Describe Saved SQL History

CD_NUMBER	CD_TITLE	YEAR	ARTIST
89	Soft Beats	2000	DJ Soft
90	Hard Rock	2000	DJ Hard

2 rows returned in 0.02 seconds Download

7. Use the read_copy_d_cds view to delete any CD of year 2000 from the underlying copy_d_cds.

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 20 Clear Command Find Tables

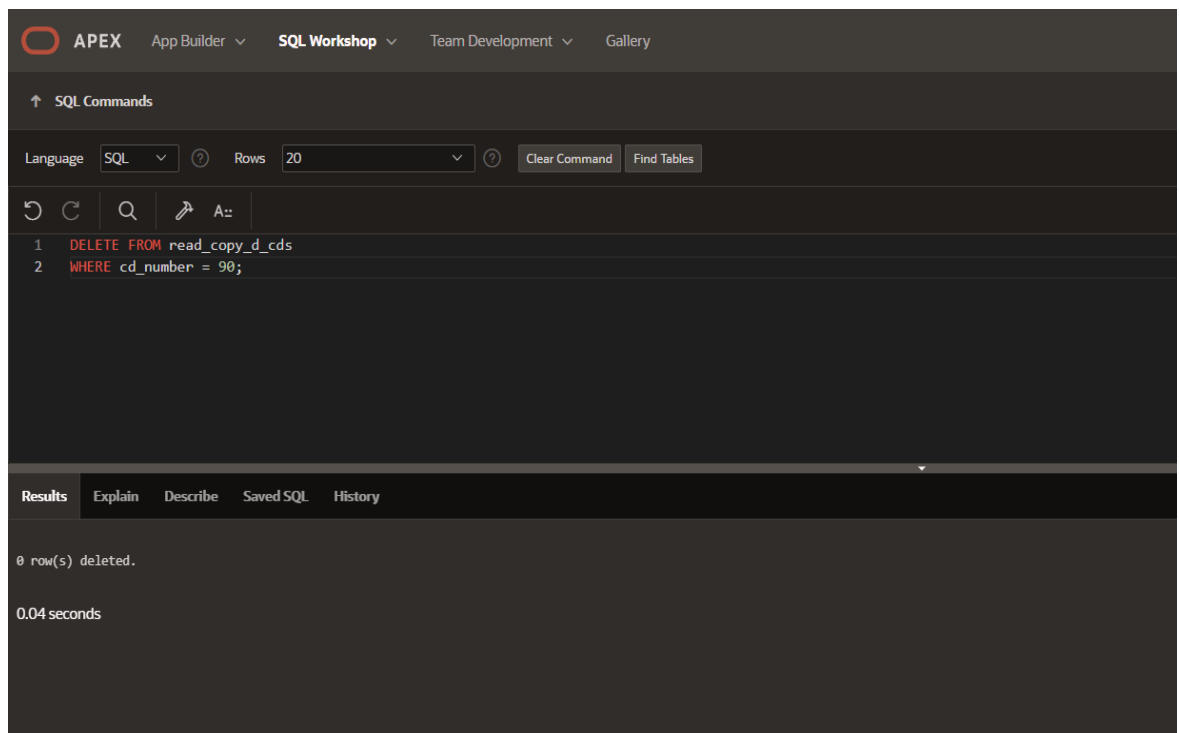
```
1 DELETE FROM read_copy_d_cds
2 WHERE year = 2000;
```

Results Explain Describe Saved SQL History

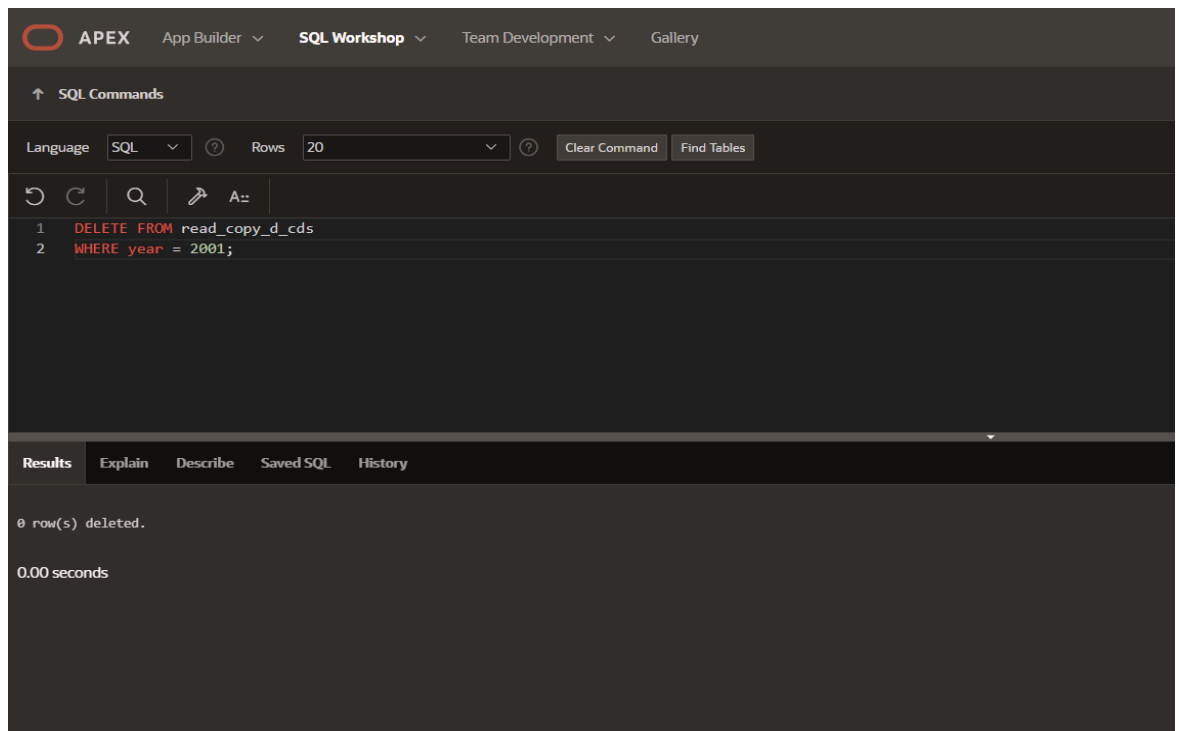
2 row(s) deleted.

0.02 seconds

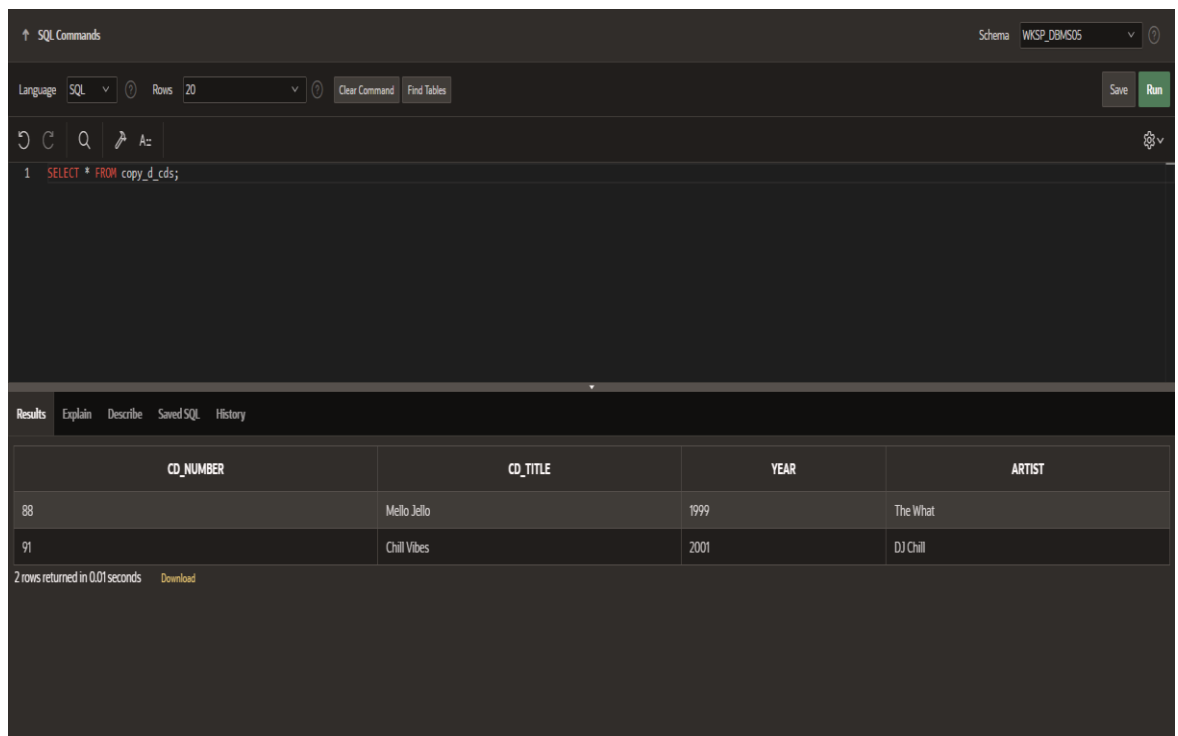
8. Use the read_copy_d_cds view to delete cd_number 90 from the underlying copy_d_cds table.



9. Use the read_copy_d_cds view to delete year 2001 records.



10. Execute a `SELECT *` statement for the base table `copy_d_cds`. What rows were deleted?



The screenshot shows a SQL IDE interface. At the top, the 'SQL Commands' tab is active, displaying the query: `1 SELECT * FROM copy_d_cds;`. The 'Schema' dropdown is set to 'WKSPP_DBMS05'. Below the query editor, the 'Results' tab is active, showing a table with 2 rows. The table has columns: CD_NUMBER, CD_TITLE, YEAR, and ARTIST. The first row has values 88, Mello Jello, 1999, and The What. The second row has values 91, Chill Vibes, 2001, and DJ Chill. Below the table, it says '2 rows returned in 0.01 seconds' and there is a 'Download' link.

CD_NUMBER	CD_TITLE	YEAR	ARTIST
88	Mello Jello	1999	The What
91	Chill Vibes	2001	DJ Chill

Result: The output shows only the rows where year \neq 2000; all rows with year = 2000 have been removed.

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

Answer:

Restrictions on modifying data through a view

When you try to INSERT, UPDATE, or DELETE data through a view, there are some restrictions:

- **Updatable View Requirement:** A view must be updatable to modify data through it. Simple views based on a single table without aggregates or joins are usually updatable.
- **Restrictions for Non-Updatable Views:** Views that include GROUP BY, DISTINCT, aggregate functions (SUM, AVG, MAX, MIN), joins, or subqueries generally cannot be directly updated.
- **WITH CHECK OPTION Restriction:** If a view has WITH CHECK OPTION, any INSERT or UPDATE through the view must satisfy the view's WHERE clause. You cannot insert or update a row that would fall outside the view's filter condition.
- **Read-Only Views:** Views created with WITH READ ONLY cannot be modified at all. Any attempt to INSERT, UPDATE, or DELETE will give an error.

- Column Restrictions: You cannot update columns that are computed expressions (like salary * 1.1) or constants in the view. Only real table columns that are included in the view can be modified.

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

Answer:

Moore's Law

Moore's Law, proposed by Gordon E. Moore in 1965, states that the number of transistors on a microchip doubles about every two years, which increases computing power and reduces costs. It has guided the growth of the semiconductor industry and technological advancements in computers.

Indefinite Continuation of Moore's Law

Moore's Law is unlikely to continue indefinitely because of physical limits of miniaturization, rising manufacturing costs, and new computing technologies like quantum and neuromorphic computing. These factors make further transistor scaling increasingly difficult.

Supporting Evidence

According to Investopedia, as transistors reach atomic scales, quantum effects and heat issues make further miniaturization difficult. McKinsey reports that rising costs and alternative computing technologies like quantum and neuromorphic computing are challenging the traditional scaling predicted by Moore's Law.

These sources support the opinion that Moore's Law cannot continue indefinitely.

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

Answer:

The Singularity in Computing

In computing, the singularity refers to a future point where artificial intelligence (AI) surpasses human intelligence, leading to rapid and unpredictable technological growth. After this point, machines could potentially improve themselves without human intervention, resulting in profound changes to society, technology, and human life.

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.

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' section is active. The 'Language' is set to 'SQL' and 'Rows' is set to '20'. The SQL command entered is:

```
1 CREATE OR REPLACE VIEW view_copy_d_songs AS
2 SELECT
3     TITLE,
4     ARTIST
5 FROM
6     COPY_D_SONGS;
```

The 'Results' tab is selected, showing the message 'View created.' and the execution time '0.06 seconds'.

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' section is active. The 'Language' is set to 'SQL' and 'Rows' is set to '20'. The SQL command entered is:

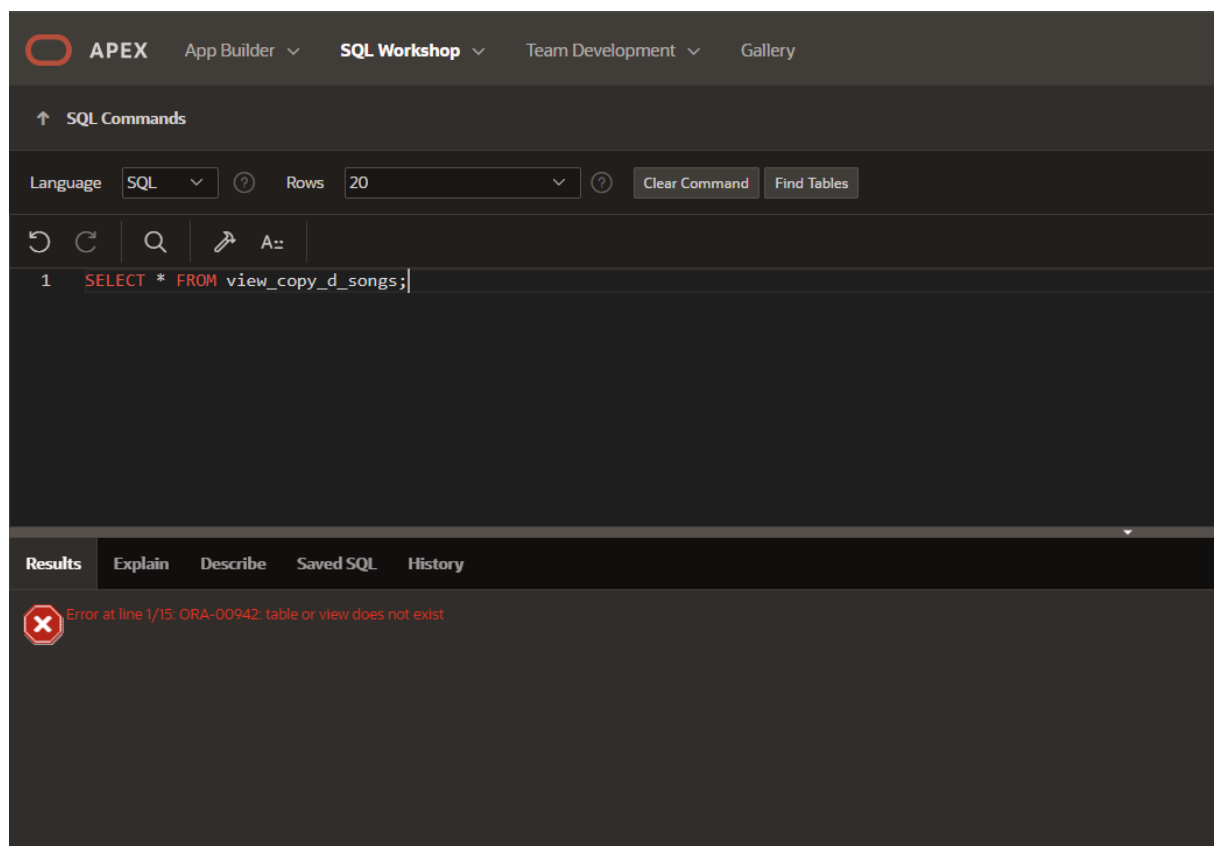
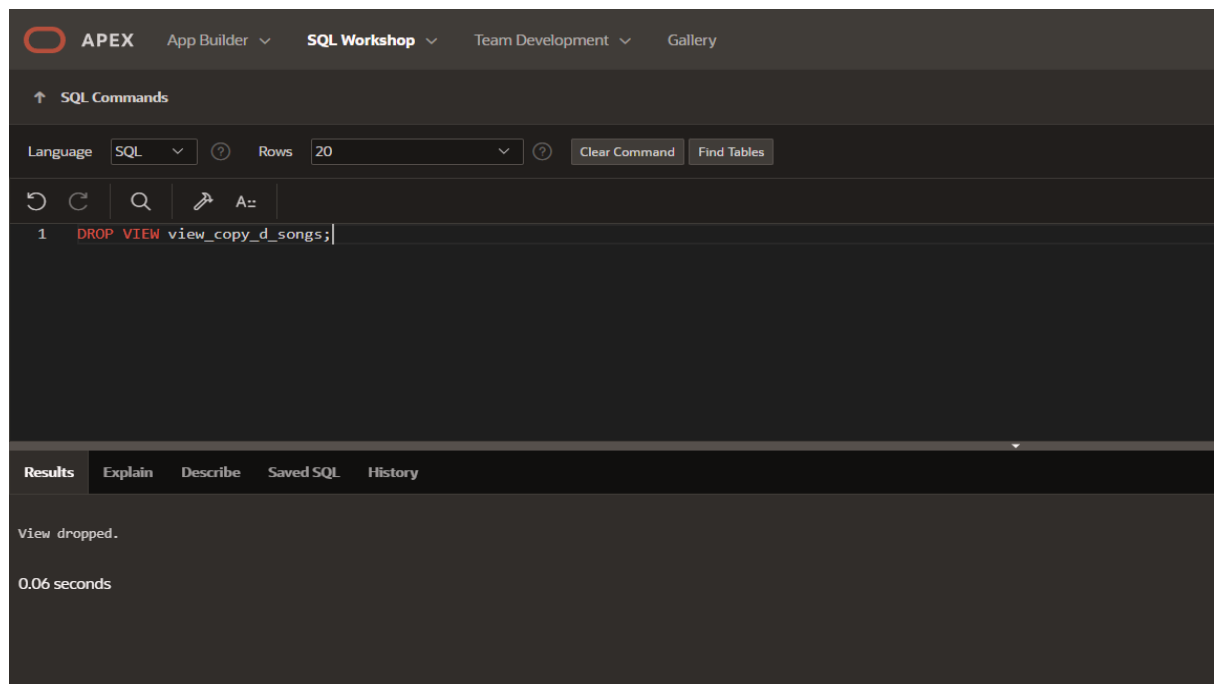
```
1 SELECT * FROM view_copy_d_songs;
```

The 'Results' tab is selected, showing the results of the query. The results are displayed in a table with two columns: 'TITLE' and 'ARTIST'.

TITLE	ARTIST
Dreamscape	DJ Dream
Happy Vibes	DJ Joy
Mello Jello	The What

3 rows returned in 0.02 seconds

2. Issue a DROP view_copy_d_songs. Execute a SELECT * statement to verify that the view has been deleted.



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.

The screenshot shows the SQL Developer interface with the following SQL query:

```
1 SELECT LAST_NAME, SALARY, SALARY_RANK
2 FROM (
3     SELECT
4         LAST_NAME,
5         SALARY,
6         RANK() OVER (ORDER BY SALARY DESC) AS SALARY_RANK
7     FROM EMPLOYEES
8 )
9 WHERE SALARY_RANK <= 3;
```

The results tab displays the following data:

LAST_NAME	SALARY	SALARY_RANK
King	24000	1
Kochhar	17000	2
De Haan	17000	2

3 rows returned in 0.02 seconds

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.

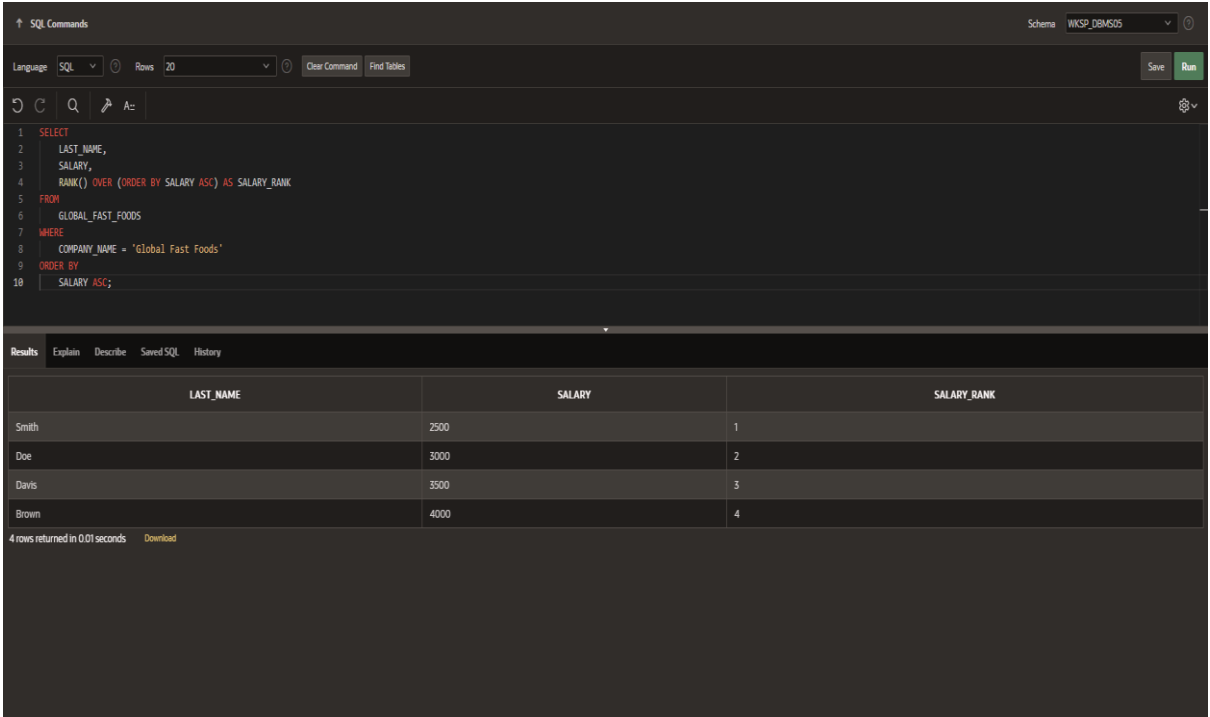
The screenshot shows the SQL Developer interface with the following SQL query:

```
1 SELECT
2     E.LAST_NAME,
3     E.SALARY,
4     E.DEPARTMENT_ID,
5     D.MAX_SALARY
6 FROM
7     EMPLOYEES E,
8     (SELECT DEPARTMENT_ID, MAX(SALARY) AS MAX_SALARY
9      FROM EMPLOYEES
10     GROUP BY DEPARTMENT_ID) D
11 WHERE
12     E.DEPARTMENT_ID = D.DEPARTMENT_ID;
```

The results tab displays the following data:

LAST_NAME	SALARY	DEPARTMENT_ID	MAX_SALARY
Brown	3500	50	3500
White	4000	90	9000
Morgan	9000	90	9000
Doe	7000	90	9000
King	24000	10	24000
Kochhar	17000	10	24000
De Haan	17000	10	24000
Ernst	6000	60	9000
Hunold	9000	60	9000
Davies	5000	80	5000
Austin	4800	80	5000

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



The screenshot shows a SQL IDE interface with a dark theme. At the top, there's a toolbar with 'SQL Commands', 'Schema' (set to 'WKS2_DBMS05'), 'Language' (set to 'SQL'), 'Rows' (set to '20'), 'Clear Command', and 'Find Tables'. Below the toolbar is a command input area with a 'Save' button and a green 'Run' button. The main area displays a SQL query:

```
1 SELECT
2     LAST_NAME,
3     SALARY,
4     RANK() OVER (ORDER BY SALARY ASC) AS SALARY_RANK
5 FROM
6     GLOBAL_FAST_FOODS
7 WHERE
8     COMPANY_NAME = 'Global Fast Foods'
9 ORDER BY
10    SALARY ASC;
```

Below the query, there's a 'Results' tab with sub-tabs 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with 4 rows:

LAST_NAME	SALARY	SALARY_RANK
Smith	2500	1
Doe	3000	2
Davis	3500	3
Brown	4000	4

At the bottom of the results section, it says '4 rows returned in 0.01 seconds' and has a 'Download' link.

Indexes and Synonyms

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

Answer:

Index in Databases

An index is a database object that improves the speed of data retrieval from a table. It works like an index in a book — instead of scanning the whole table, the database uses the index to quickly locate the rows that match a query.

Uses of an index:

Faster search: Speeds up SELECT queries by reducing the amount of data scanned.

Efficient sorting: Helps with ORDER BY and GROUP BY operations.

Uniqueness enforcement: A unique index ensures that no two rows have the same value in the indexed column.

Quick joins: Improves performance when joining large tables.

Example:

```
CREATE INDEX idx_employee_lastname  
ON EMPLOYEES(LAST_NAME);
```

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

Answer:

ROWID in Databases

A ROWID is a unique identifier for each row in an Oracle database table. It represents the physical location of the row on disk (datafile, block, and row position).

Uses of ROWID:

Fast access: Queries using ROWID are very fast because the database knows the exact physical location of the row.

Updating or deleting specific rows: You can use ROWID to precisely update or delete a row.

Uniqueness check: Even if a table has no primary key, each row has a unique ROWID.

Example:

```
SELECT ROWID, FIRST_NAME, LAST_NAME  
FROM EMPLOYEES  
WHERE LAST_NAME = 'King';
```

3. When will an index be created automatically?

Automatic Index Creation

In Oracle (and most relational databases), an index is automatically created in the following situations:

Primary Key Constraint: When you define a column as a PRIMARY KEY, Oracle automatically creates a unique index on that column to enforce uniqueness.

Unique Key Constraint: When you define a column as a UNIQUE, Oracle automatically creates a unique index to ensure no duplicate values are allowed.

Indexes are not automatically created for normal columns unless specified. You must explicitly create them for faster searches on non-key columns.

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.

The screenshot shows the Oracle APEX SQL Workshop interface. The 'SQL Commands' tab is active, displaying the following SQL code:

```
1 CREATE INDEX idx_cd_number
2 ON D_TRACK_LISTINGS(cd_number);
```

Below the code editor, the 'Results' tab shows the message 'Index created.' and the execution time '0.04 seconds'.

The screenshot shows the Oracle APEX SQL Workshop interface with the 'SQL Commands' tab active. The SQL code entered is:

```
1 SELECT index_name, table_name, uniqueness
2 FROM user_indexes
3 WHERE table_name = 'D_TRACK_LISTINGS';
```

The 'Results' tab displays the following table:

INDEX_NAME	TABLE_NAME	UNIQUENESS
IDX_CD_NUMBER	D_TRACK_LISTINGS	NONUNIQUE
SYS_C00187104669	D_TRACK_LISTINGS	UNIQUE

At the bottom, it states '2 rows returned in 0.09 seconds' with a 'Download' link.

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.

The screenshot shows the SQL Developer interface with the following SQL query:

```
1 SELECT
2   ui.index_name,
3   ui.table_name,
4   ui.uniqueness,
5   uic.column_name
6 FROM
7   user_indexes ui
8 JOIN
9   user_ind_columns uic
10 ON
11   ui.index_name = uic.index_name
12 WHERE
13   ui.table_name = 'D_SONGS';
```

The results are displayed in a table with the following columns: INDEX_NAME, TABLE_NAME, UNIQUENESS, and COLUMN_NAME.

INDEX_NAME	TABLE_NAME	UNIQUENESS	COLUMN_NAME
IDX_D_SONGS_ARTIST	D_SONGS	NONUNIQUE	ARTIST
SYS_C00187127059	D_SONGS	UNIQUE	ID

2 rows returned in 0.30 seconds

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.

The screenshot shows the SQL Developer interface with the following SQL query:

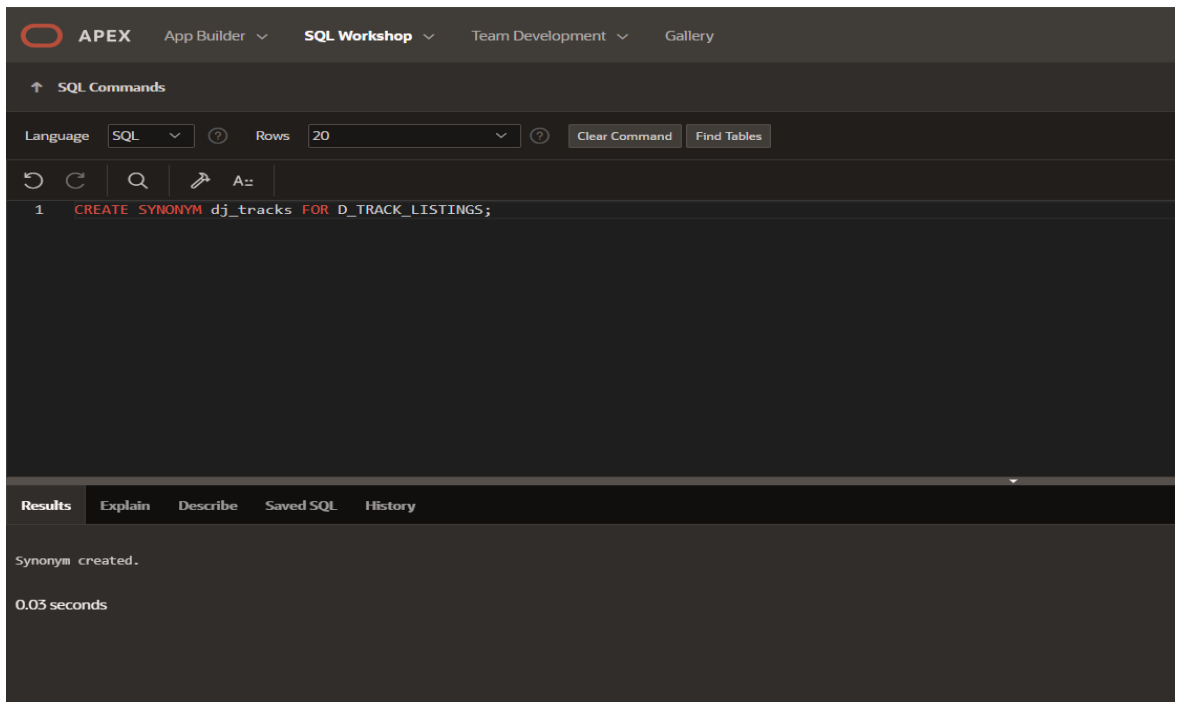
```
1 SELECT
2   index_name,
3   table_name,
4   uniqueness
5 FROM
6   user_indexes
7 WHERE
8   table_name = 'D_EVENTS';
```

The results are displayed in a table with the following columns: INDEX_NAME, TABLE_NAME, and UNIQUENESS.

INDEX_NAME	TABLE_NAME	UNIQUENESS
IDX_D_EVENTS_EVENT_DATE	D_EVENTS	NONUNIQUE

1 rows returned in 0.07 seconds

7. Write a query to create a synonym called dj_tracks for the DJs on Demand d_track_listings table.

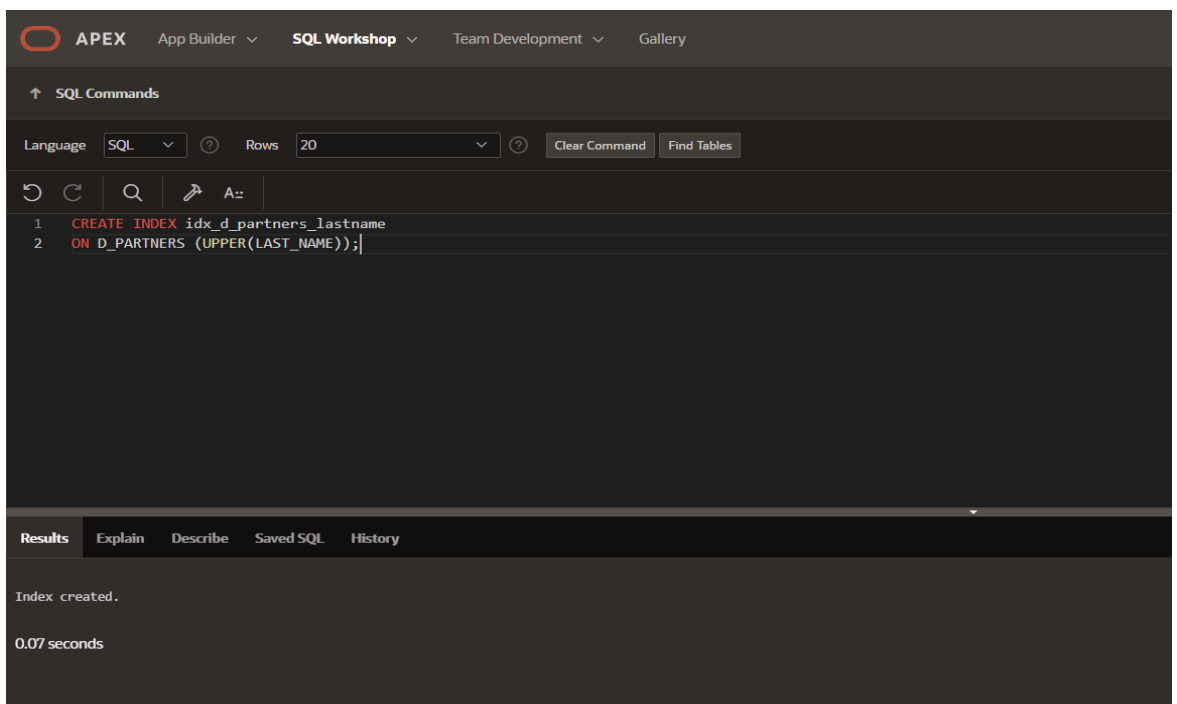


The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' section is active. The 'Language' dropdown is set to 'SQL', and the 'Rows' dropdown is set to '20'. The 'Clear Command' and 'Find Tables' buttons are visible. The SQL command area contains the following text:

```
1 CREATE SYNONYM dj_tracks FOR D_TRACK_LISTINGS;
```

The 'Results' tab is selected, showing the message 'Synonym created.' and the execution time '0.03 seconds'.

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.



The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. Below the navigation bar, the 'SQL Commands' section is active. The 'Language' dropdown is set to 'SQL', and the 'Rows' dropdown is set to '20'. The 'Clear Command' and 'Find Tables' buttons are visible. The SQL command area contains the following text:

```
1 CREATE INDEX idx_d_partners_lastname  
2 ON D_PARTNERS (UPPER(LAST_NAME));
```

The 'Results' tab is selected, showing the message 'Index created.' and the execution time '0.07 seconds'.

SQL Commands

Schema: WKSP_DBMS05

Language: SQL Rows: 20 Clear Command Find Tables Save Run

```
1 SELECT *
2 FROM D_PARTNERS
3 WHERE UPPER(LAST_NAME) = 'SMITH';
```

Results Explain Describe Saved SQL History

PARTNER_ID	LAST_NAME	FIRST_NAME
2	Smith	Alice

1 rows returned in 0.01 seconds Download

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

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 20 Clear Command Find Tables

```
1 CREATE SYNONYM dj_tracks FOR D_TRACK_LISTINGS;
```

Results Explain Describe Saved SQL History

Synonym created.

0.03 seconds

SQL Commands

Schema: WKSP_DBMS05

Language: SQL Rows: 20

Clear Command Find Tables Save Run

```
1 SELECT synonym_name, table_name, table_owner
2 FROM user_synonyms
3 WHERE table_name = 'D_TRACK_LISTINGS';
```

Results Explain Describe Saved SQL History

SYNONYM_NAME	TABLE_NAME	TABLE_OWNER
DJ_TRACKS	D_TRACK_LISTINGS	WKSP_DBMS05

1 rows returned in 0.05 seconds Download

10. Drop the synonym that you created in question.

APEX SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 20

Clear Command Find Tables

```
1 DROP SYNONYM dj_tracks;
```

Results Explain Describe Saved SQL History

Synonym dropped.

0.03 seconds