

EXERCISE-13

Creating Views

Name: VIDHYASHREE J

Register Number:

241801310 Department:

AI & DS

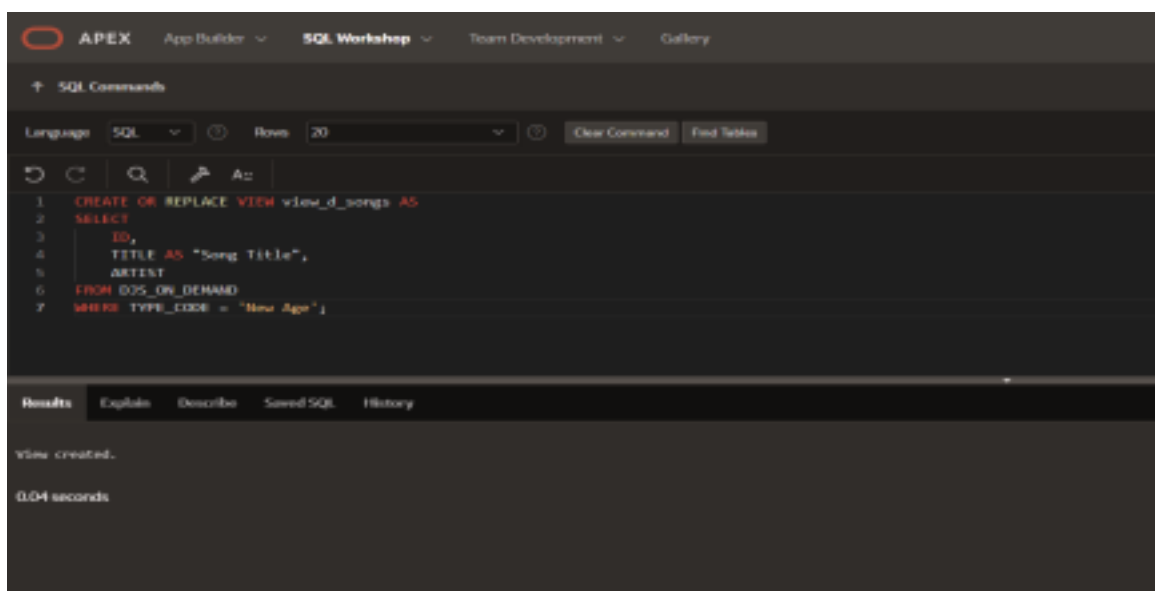
1. What are three uses for a view from a DBA's perspective?

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 APEX SQL Workshop interface. The SQL Commands pane contains the following code:

```
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';
```

The Results pane shows 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 following query and results:

```
SELECT * FROM view_d_songs
```

ID	Song Title	ARTIST
1	Cake Walk	Ed Green
3	Myxie Med	Ed Zim

7 rows returned in 0.07 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 following query and results:

```
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 DIS_ON_DEMAND
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 SQL Commands tab is active, displaying the following SQL 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 shows 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 SQL Commands tab is active, displaying the following SQL 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 shows the message "View created." and the execution time "0.05 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.

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 20 Clear Command Find Tables

1 DESC COPY_D_SONGS;

Results Explain Describe Saved SQL History

Object Type: TABLE Object: COPY_D_SONGS

Table	Columns	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	-	-	-	✓	-	-

APEX App Builder SQL Workshop Team Development Gallery

SQL Commands

Language: SQL Rows: 20 Clear Command Find Tables

1 DESC COPY_D_EVENTS;

Results Explain Describe Saved SQL History

Object Type: TABLE Object: COPY_D_EVENTS

Table	Columns	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

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

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.

SQL Commands

Language: SQL | Run | Clear Command | Find Table | Save | Run

```

1 SELECT TABLE_NAME, COLUMN_NAME, UPDATABLE, INSERTABLE, DELETABLE
2 FROM INFORMATION_SCHEMA.COLUMNS
3 WHERE TABLE_NAME = 'COPY_0_SONGS';

```

Results | Explain | Describe | Saved SQL | History

TABLE_NAME	COLUMN_NAME	UPDATABLE	INSERTABLE	DELETABLE
COPY_0_SONGS	ID	YES	YES	YES
COPY_0_SONGS	TITLE	YES	YES	YES
COPY_0_SONGS	DURATION	YES	YES	YES
COPY_0_SONGS	ARTIST	YES	YES	YES
COPY_0_SONGS	TYPE_CODE	YES	YES	YES

1 rows returned in 0.07 seconds | Download

SQL Commands

Language: SQL | Run | Clear Command | Find Table | Save | Run

```

1 SELECT TABLE_NAME, COLUMN_NAME, UPDATABLE, INSERTABLE, DELETABLE
2 FROM INFORMATION_SCHEMA.COLUMNS
3 WHERE TABLE_NAME = 'COPY_0_EVENTS';

```

Results | Explain | Describe | Saved SQL | History

TABLE_NAME	COLUMN_NAME	UPDATABLE	INSERTABLE	DELETABLE
COPY_0_EVENTS	EVENT_ID	YES	YES	YES
COPY_0_EVENTS	EVENT_NAME	YES	YES	YES
COPY_0_EVENTS	EVENT_DATE	YES	YES	YES
COPY_0_EVENTS	THEME_DESCRIPTION	YES	YES	YES
COPY_0_EVENTS	RACE	YES	YES	YES

1 rows returned in 0.06 seconds | Download

SQL Commands

Language: SQL | Run | Clear Command | Find Table | Save | Run

```

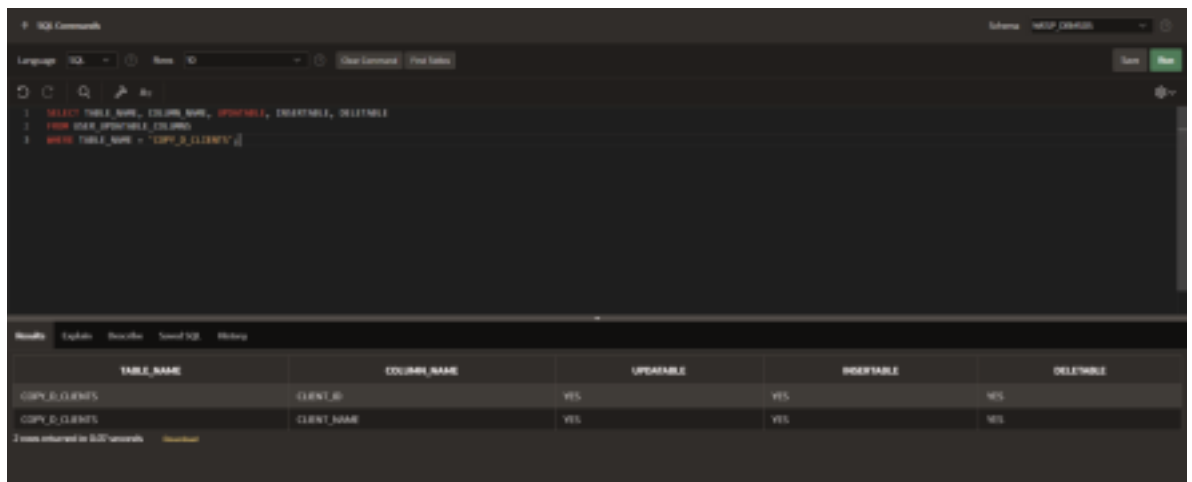
1 SELECT TABLE_NAME, COLUMN_NAME, UPDATABLE, INSERTABLE, DELETABLE
2 FROM INFORMATION_SCHEMA.COLUMNS
3 WHERE TABLE_NAME = 'COPY_0_CDS';

```

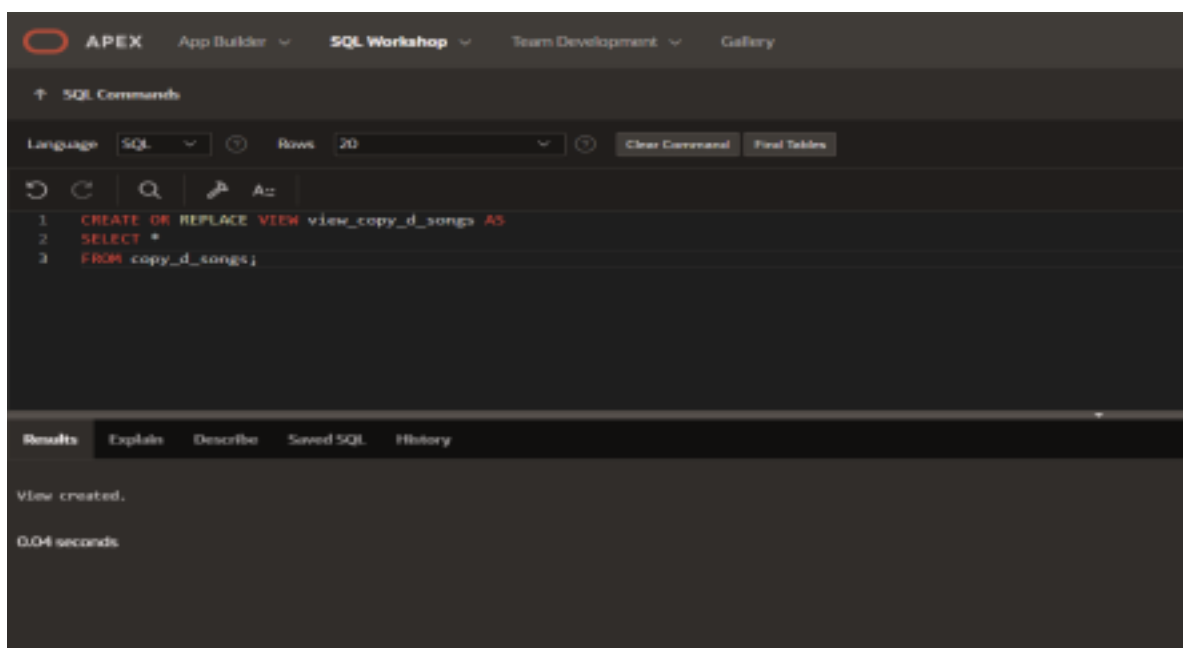
Results | Explain | Describe | Saved SQL | History

TABLE_NAME	COLUMN_NAME	UPDATABLE	INSERTABLE	DELETABLE
COPY_0_CDS	CD_NUMBER	YES	YES	YES
COPY_0_CDS	CD_TITLE	YES	YES	YES
COPY_0_CDS	YEAR	YES	YES	YES
COPY_0_CDS	ARTIST	YES	YES	YES

4 rows returned in 0.06 seconds | Download

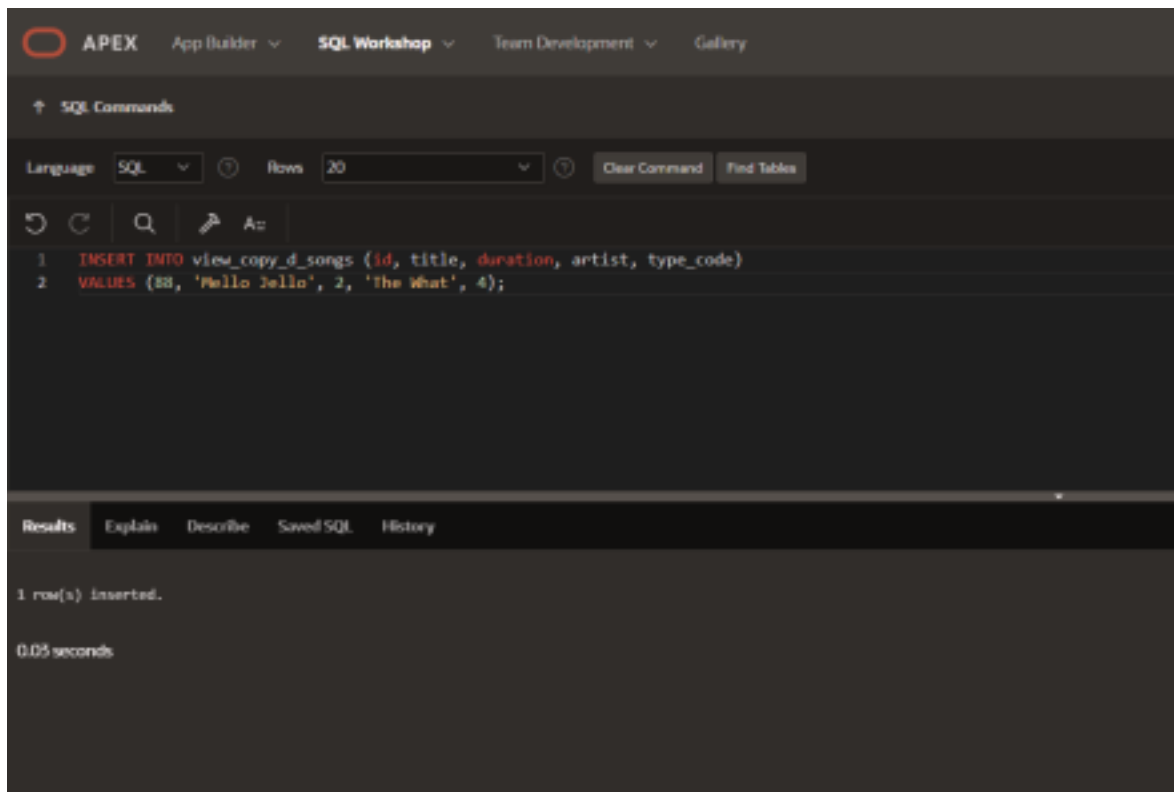


- Use the CREATE or REPLACE option to create a view of all the columns in the copy_d_songs table called view_copy_d_songs.



- 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



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.



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.





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



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?



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.



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.



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.



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



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.





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.



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.



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



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.



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



10. Drop the synonym that you created in question.

