

Database Programming with SQL 16-2: Indexes and Synonyms Practice Activities

# Objectives

* Define an index and its use as a schema object
* Name the conditions that cause an index to be created automatically
* Create and execute a CREATE INDEX and DROP INDEX statement
* Construct and execute a function-based index
* Construct a private and public synonym

# Vocabulary

Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| Confirming Index | Confirms the existence of indexes from the USER\_INDEXES data dictionary view |
| **INDEX** | Schema object that speeds up retrieval of rows |
| **CREATE PUBLIC SYNONYM** | To refer to a table by another name to simplify access |
| **Composite Index** | An index that you create on multiple columns in a table |
| **Unique index** | The Oracle Server automatically creates this index when you define a column in a table to have a PRIMARY KEY or a UNIQUE KEY constraint |
| **function-based index** | Stores the indexed values and uses the index based on a SELECT statement to retrieve the data |
| **DROP INDEX** | Removes an index |
| **Synonym** | Gives alternative names to objects |

# Try It / Solve It

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

это объекты схемы, которые ускоряют извлечение строк из таблицы.

- Они предназначены для эффективного поиска данных в базе данных. Я могу удалить индекс, если запросы в приложении не используют какой-либо индекс или говорят, что это не ускоряет запросы или может быть таблица очень маленькая. Индекс обеспечивает прямой и быстрый доступ к строке в таблице.

· Я должен создать индекс, если таблица большая и ожидается, что большинство запросов будут извлекать менее 2–4 процентов строк.

· Я должен создать индекс, если один или несколько столбцов часто используются вместе в условии соединения.

Назначение : Индекс обеспечивает прямой и быстрый доступ к строке в таблице. Они обеспечивают индексированный путь для быстрого поиска данных, тем самым уменьшая необходимость тяжелых операций ввода-вывода на диск.

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

**Индексы используют ROWID (строковое представление адреса строки с основанием 64, содержащее идентификатор блока, местоположение строки в блоке и идентификатор файла базы данных), что является самым быстрым способом доступа к любой конкретной строке.**

1. When will an index be created automatically?

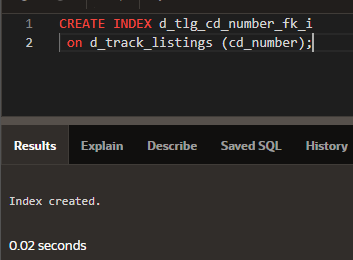
**Для первичных/уникальных ключей: Хотя уникальный индекс можно создать вручную, предпочтительно использовать ограничение уникальности/первичности в таблице. Таким образом, это означает, что первичный ключ/уникальный ключ использует уже существующий уникальный индекс, но если индекс еще не существует, он создается при применении ограничения уникальности/первичного ключа.**

**Oracle также автоматически создает индекс для хранения больших объектов, типа xml и материализованного представления.**

1. Create a nonunique index (foreign key) for the DJs on Demand column (cd\_number) in the D\_TRACK\_LISTINGS table. Use the Oracle Application Developer SQL Workshop Data Browser to confirm that the index was created.

CREATE INDEX d\_tlg\_cd\_number\_fk\_i

on d\_track\_listings (cd\_number);

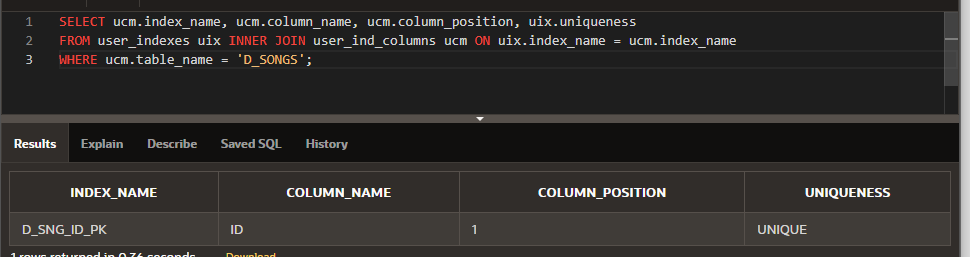


1. Use the join statement to display the indexes and uniqueness that exist in the data dictionary for the DJs on Demand D\_SONGS table.

SELECT ucm.index\_name, ucm.column\_name, ucm.column\_position, uix.uniqueness

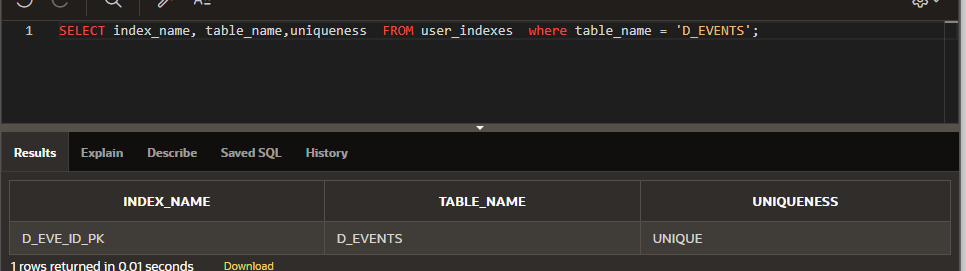
FROM user\_indexes uix INNER JOIN user\_ind\_columns ucm ON uix.index\_name = ucm.index\_name

WHERE ucm.table\_name = 'D\_SONGS';



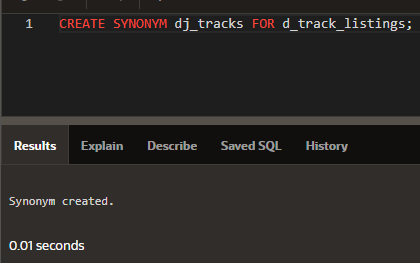
1. Use a SELECT statement to display the index\_name, table\_name, and uniqueness from the data dictionary USER\_INDEXES for the DJs on Demand D\_EVENTS table.

SELECT index\_name, table\_name,uniqueness FROM user\_indexes where table\_name = 'D\_EVENTS';

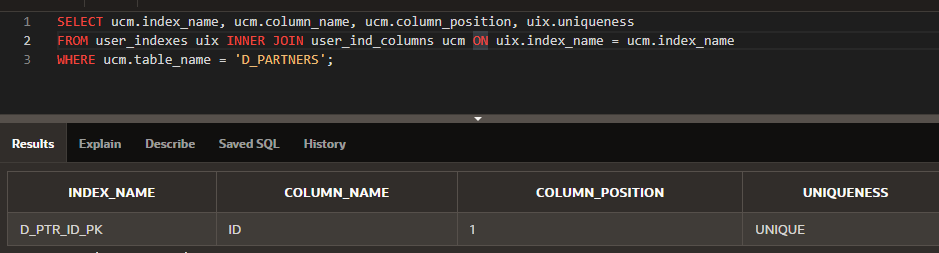


1. Write a query to create a synonym called dj\_tracks for the DJs on Demand d\_track\_listings table.

CREATE SYNONYM dj\_tracks FOR d\_track\_listings;



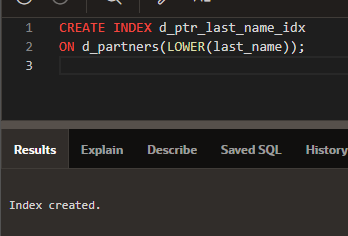
1. Create a function-based index for the last\_name column in DJs on Demand D\_PARTNERS table that makes it possible not to have to capitalize the table name for searches. Write a SELECT statement that would use this index.



SELECT ucm.index\_name, ucm.column\_name, ucm.column\_position, uix.uniqueness

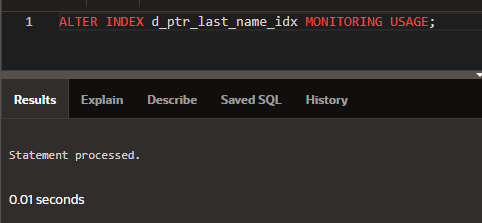
FROM user\_indexes uix INNER JOIN user\_ind\_columns ucm ON uix.index\_name = ucm.index\_name

WHERE ucm.table\_name = 'D\_PARTNERS';

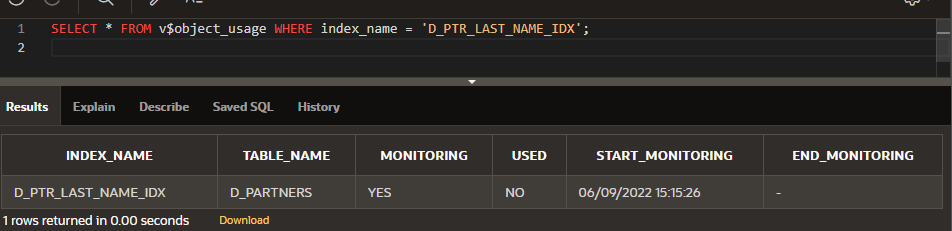


CREATE INDEX d\_ptr\_last\_name\_idx

ON d\_partners(LOWER(last\_name));



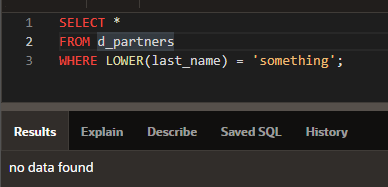
SELECT \* FROM v$object\_usage WHERE index\_name = 'D\_PTR\_LAST\_NAME\_IDX';



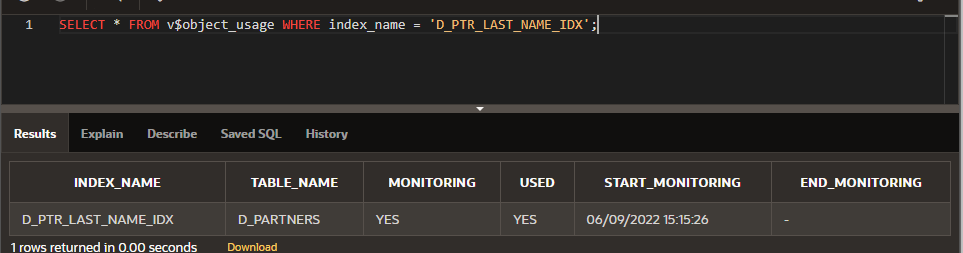
SELECT \*

FROM d\_partners

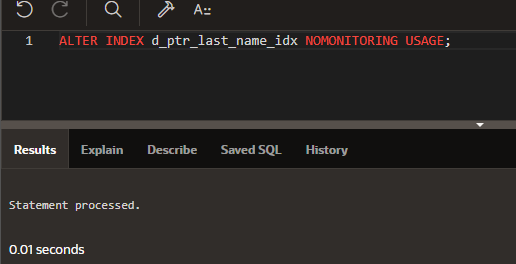
WHERE LOWER(last\_name) = 'something';



SELECT \* FROM v$object\_usage WHERE index\_name = 'D\_PTR\_LAST\_NAME\_IDX';



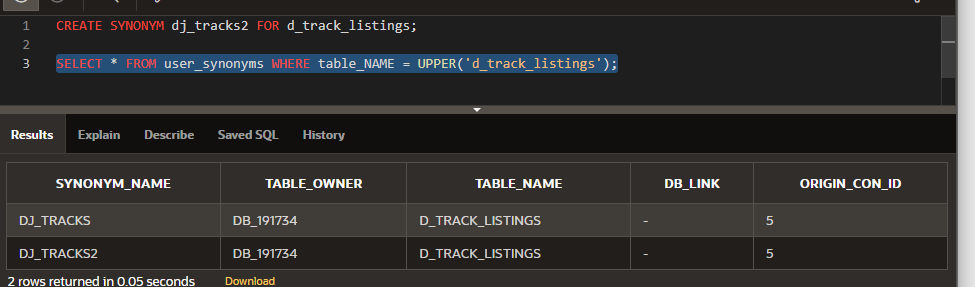
ALTER INDEX d\_ptr\_last\_name\_idx NOMONITORING USAGE;



1. Create a synonym for the D\_TRACK\_LISTINGS table. Confirm that it has been created by querying the data dictionary.

CREATE SYNONYM dj\_tracks2 FOR d\_track\_listings;

SELECT \* FROM user\_synonyms WHERE table\_NAME = UPPER('d\_track\_listings');



1. Drop the synonym that you created in question 9.

DROP SYNONYM dj\_tracks2;

SELECT \* FROM user\_synonyms WHERE table\_NAME = UPPER('d\_track\_listings');

