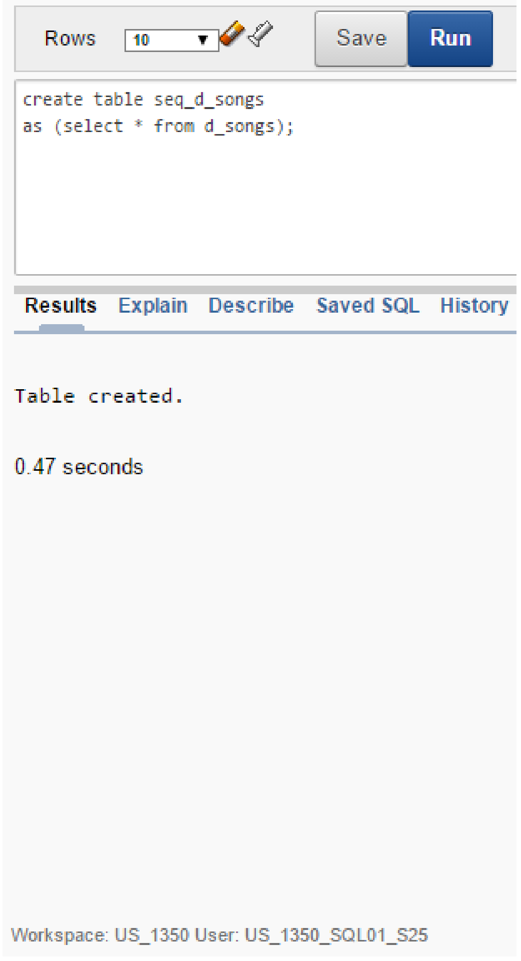
Section 12 Lesson 1: Working with Sequences

Try It / Solve It

1. Using CREATE TABLE AS subquery syntax, create a seq_d_songs table of all the columns in the DJs on Demand database table d_songs. Use the SELECT * in the subquery to make sure that you have copied all of the columns.



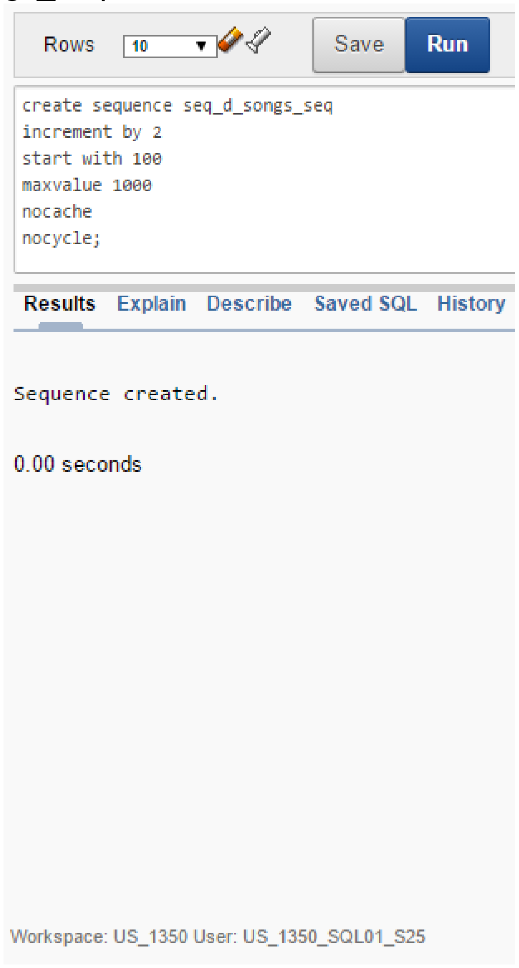
```
create table seq_d_songs
as (select * from d_songs);
```

```
Table created.

0.47 seconds
```

Workspace: US_1350 User: US_1350_SQL01_S25

2. Because you are using copies of the original tables, the only constraints that were carried over were the NOT NULL constraints. Create a sequence to be used with the primary-key column of the seq_d_songs table. To avoid assigning primary-key numbers to these tables that already exist, the sequence should start at 100 and have a maximum value of 1000. Have your sequence increment by 2 and have NOCACHE and NOCYCLE. Name the sequence seq_d_songs_seq.



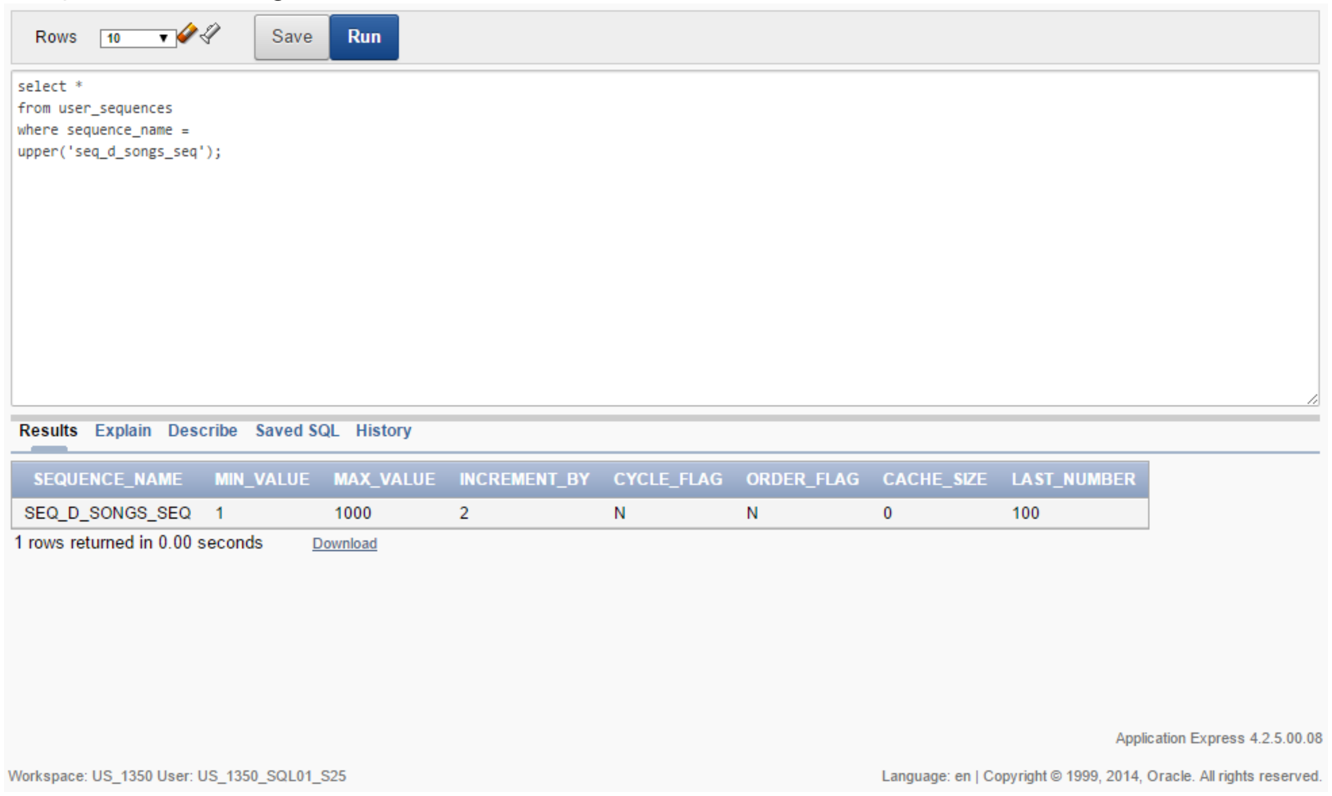
```
create sequence seq_d_songs_seq
increment by 2
start with 100
maxvalue 1000
nocache
nocycle;
```

```
Sequence created.
```

0.00 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

3. Query the USER_SEQUENCES data dictionary to verify the seq_d_songs_seq SEQUENCE settings.



```
select *
from user_sequences
where sequence_name =
upper('seq_d_songs_seq');
```

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER
SEQ_D_SONGS_SEQ	1	1000	2	N	N	0	100

1 rows returned in 0.00 seconds

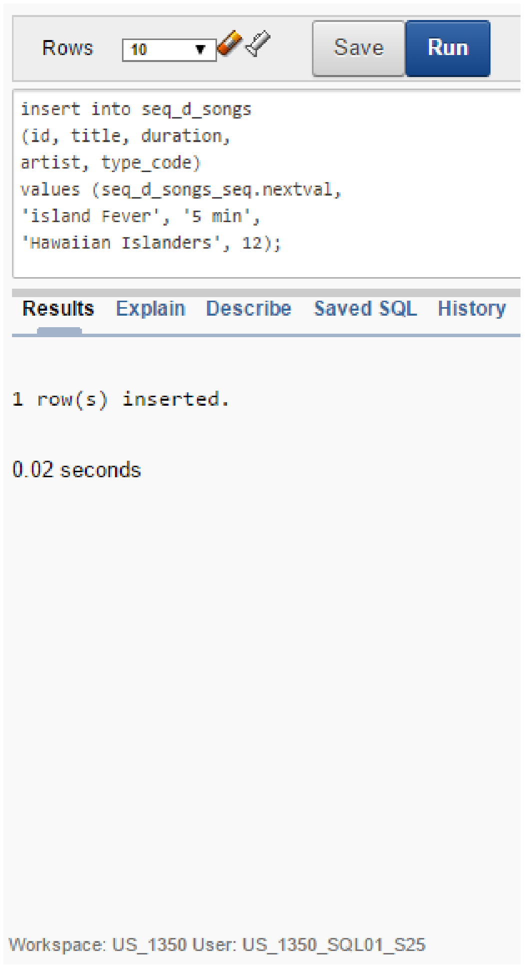
Application Express 4.2.5.00.08

Workspace: US_1350 User: US_1350_SQL01_S25

Language: en | Copyright © 1999, 2014, Oracle. All rights reserved.

4. Insert two rows into the seq_d_songs table. Be sure to use the sequence that you created for the ID column. Add the two songs shown in the graphic.

ID	TITLE	DURATION	ARTIST	TYPE_CODE
	Island Fever	5 min	Hawaiian Islanders	12
	Castle of Dreams	4 min	The Wanderers	77

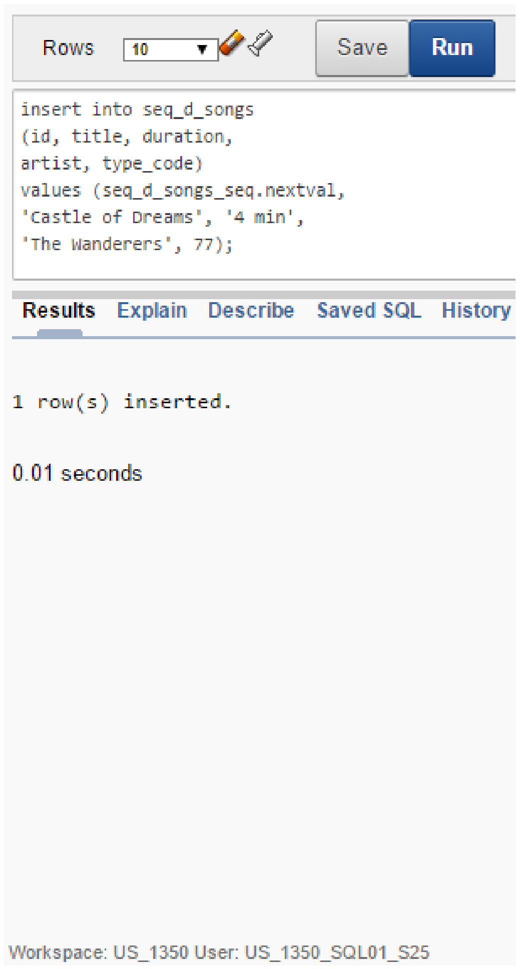


```
insert into seq_d_songs
(id, title, duration,
artist, type_code)
values (seq_d_songs_seq.nextval,
'island Fever', '5 min',
'Hawaiian Islanders', 12);
```

```
1 row(s) inserted.

0.02 seconds
```

Workspace: US_1350 User: US_1350_SQL01_S25



```
insert into seq_d_songs
(id, title, duration,
artist, type_code)
values (seq_d_songs_seq.nextval,
'Castle of Dreams', '4 min',
'The Wanderers', 77);
```

```
1 row(s) inserted.

0.01 seconds
```

Workspace: US_1350 User: US_1350_SQL01_S25

5. Write out the syntax for seq_d_songs_seq to view the current value for the sequence. Use the DUAL table. (Oracle Application Developer will not run this query.)

   SELECT seq_d_songs_seq.CURRVAL
   FROM dual;

6. What are three benefits of using SEQUENCEs?
   Sequences are time-saving, sharable, and eliminate the worry of duplicating values.

7. What are the advantages of caching sequence values?
   Cache sequences in memory provide faster access to sequence values and are populated the first time the sequence is referred to.

8. Name three reasons why gaps may occur in a sequence?
   Rolling back a statement containing a sequence, a system crash, or the same sequence being used on multiple tables.

ORACLE ACADEMY

Section 12 Lesson 2: Indexes and Synonyms

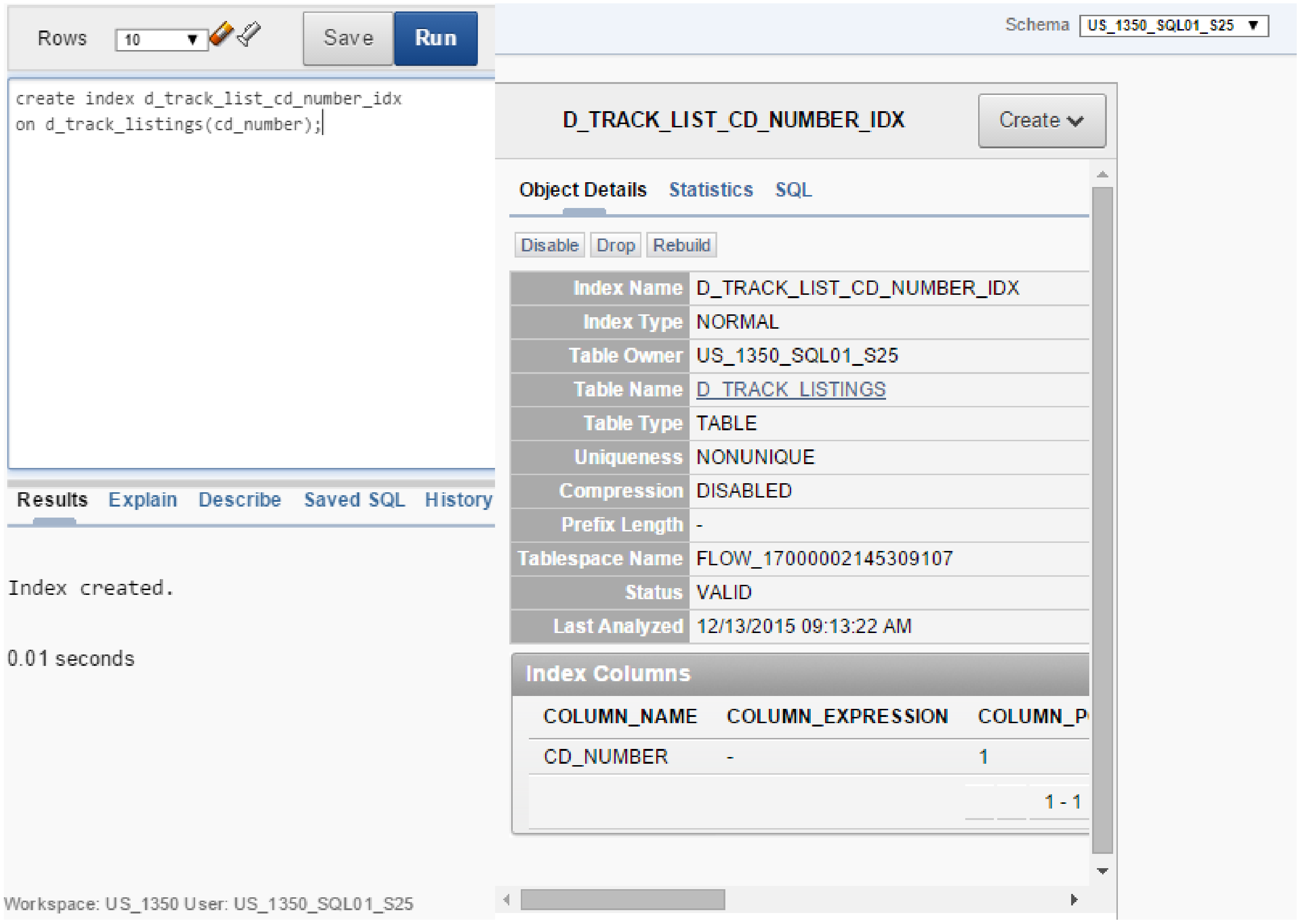
Try It / Solve It

1. What is an index and what is it used for? An index is a schema object that can speed up retrieval of rows by using a pointer. It can be used to provide direct and fast access to rows in a table.

2. What is a ROWID, and how is it used? A ROWID is a base 64 string representation of the row address containing block identifier, row location in the block, and the database file identifier.

3. When will an index be created automatically? An index will be created automatically when a column in a table is defined to have a PRIMARY KEY or UNIQUE KEY

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 Developer SQL Workshop Data Browser to confirm that the index was created.


Rows 10
Save Run
Schema US_1350_SQL01_S25
Results Explain Describe Saved SQL History
Create
Object Details Statistics SQL
Disable Drop Rebuild

```
create index d_track_list_cd_number_idx
on d_track_listings(cd_number);
```

```
Index created.

0.01 seconds
```

Workspace: US_1350 User: US_1350_SQL01_S25

D_TRACK_LIST_CD_NUMBER_IDX

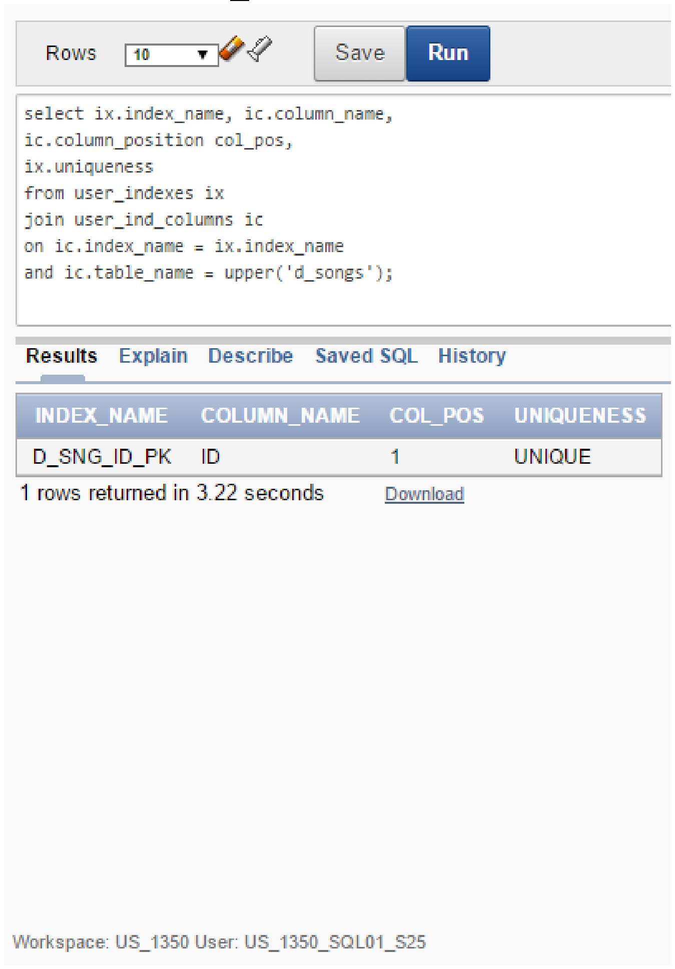
Index Name	D_TRACK_LIST_CD_NUMBER_IDX
Index Type	NORMAL
Table Owner	US_1350_SQL01_S25
Table Name	D_TRACK_LISTINGS
Table Type	TABLE
Uniqueness	NONUNIQUE
Compression	DISABLED
Prefix Length	-
Tablespace Name	FLOW_17000002145309107
Status	VALID
Last Analyzed	12/13/2015 09:13:22 AM

Index Columns

COLUMN_NAME	COLUMN_EXPRESSION	COLUMN_P
CD_NUMBER	-	1

1 - 1

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.



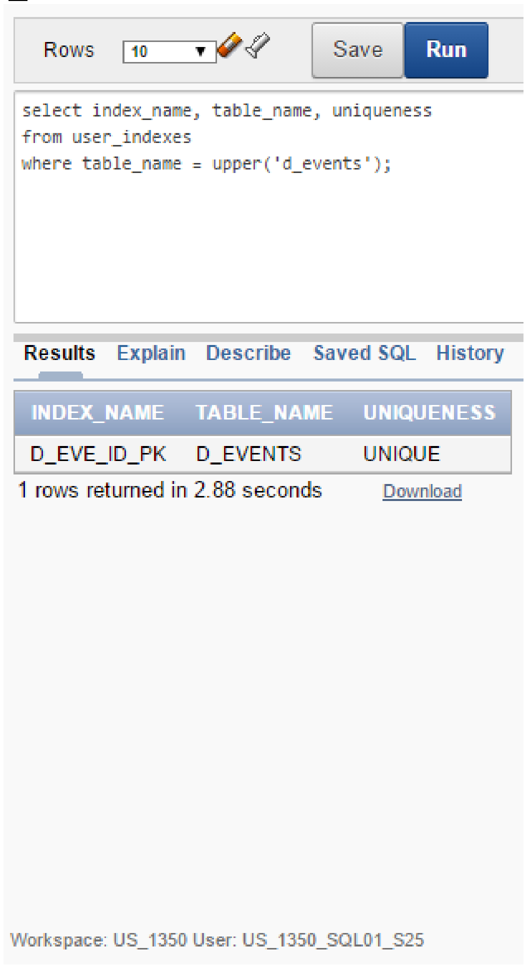
```
select ix.index_name, ic.column_name,
ic.column_position col_pos,
ix.uniqueness
from user_indexes ix
join user_ind_columns ic
on ic.index_name = ix.index_name
and ic.table_name = upper('d_songs');
```

INDEX_NAME	COLUMN_NAME	COL_POS	UNIQUENESS
D_SNG_ID_PK	ID	1	UNIQUE

1 rows returned in 3.22 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

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.



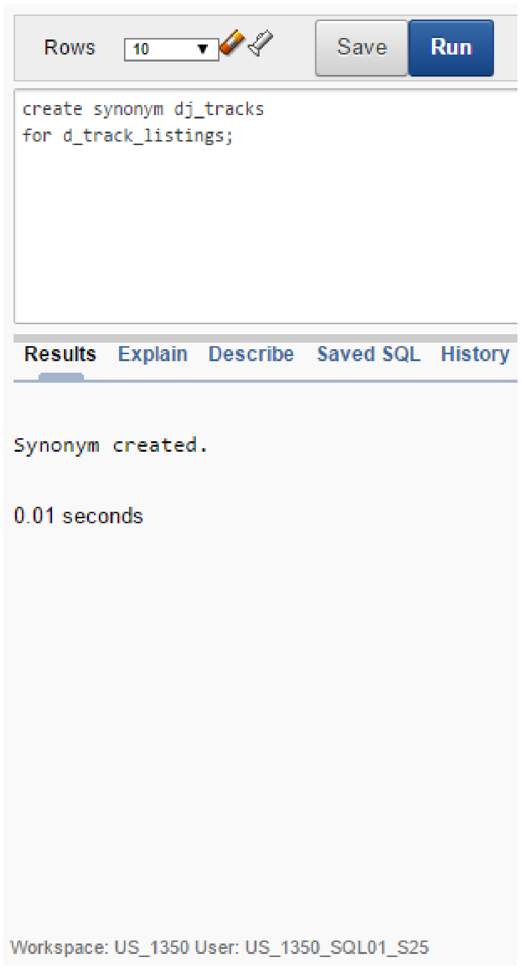
```
select index_name, table_name, uniqueness
from user_indexes
where table_name = upper('d_events');
```

INDEX_NAME	TABLE_NAME	UNIQUENESS
D_EVE_ID_PK	D_EVENTS	UNIQUE

1 rows returned in 2.88 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

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



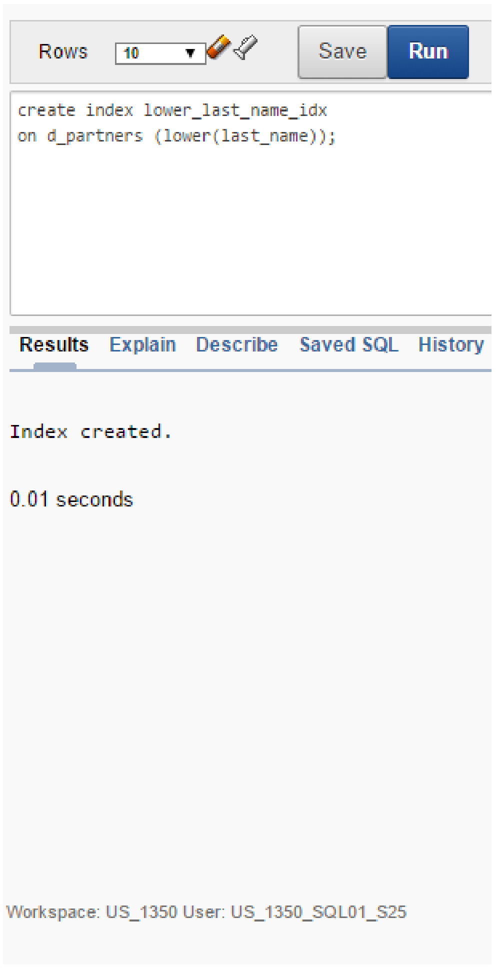
```
create synonym dj_tracks
for d_track_listings;
```

```
Synonym created.

0.01 seconds
```

Workspace: US_1350 User: US_1350_SQL01_S25

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.

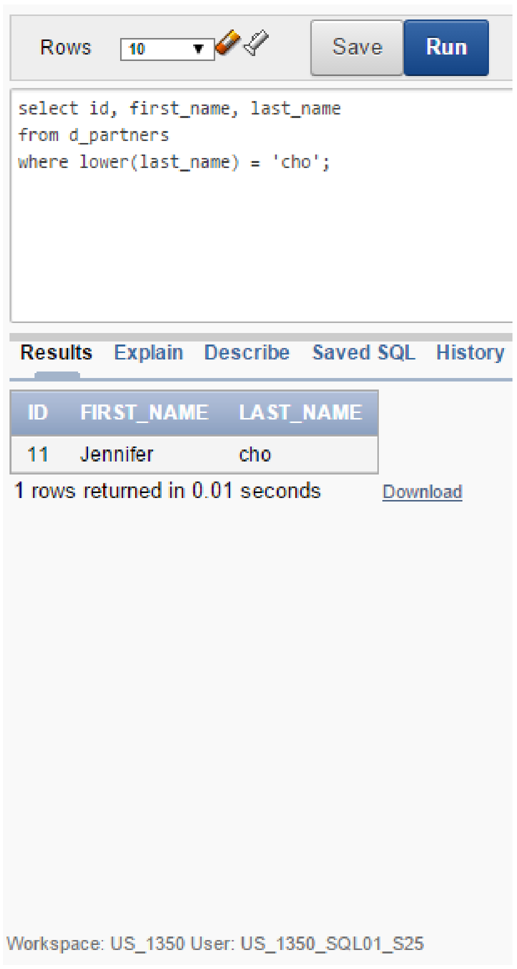


```
create index lower_last_name_idx
on d_partners (lower(last_name));
```

```
Index created.

0.01 seconds
```

Workspace: US_1350 User: US_1350_SQL01_S25



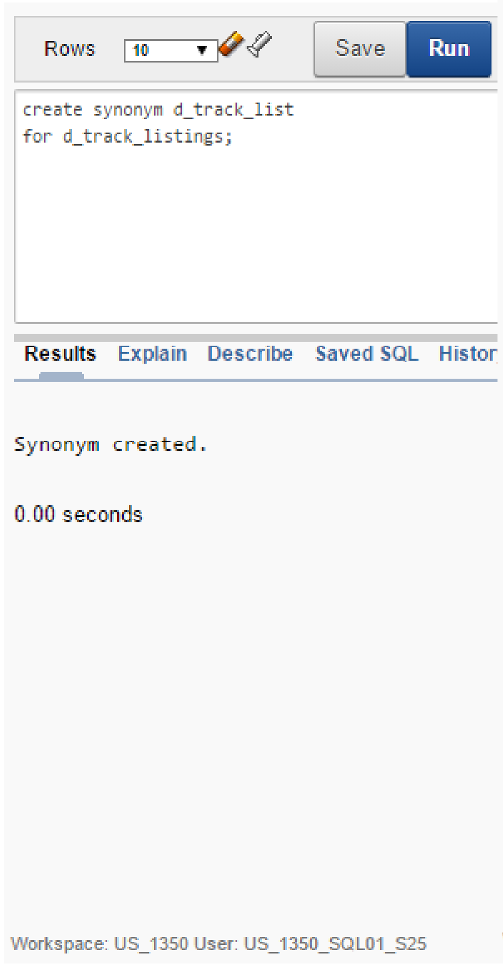
```
select id, first_name, last_name
from d_partners
where lower(last_name) = 'cho';
```

ID	FIRST_NAME	LAST_NAME
11	Jennifer	cho

1 rows returned in 0.01 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

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



Rows 10 Save Run

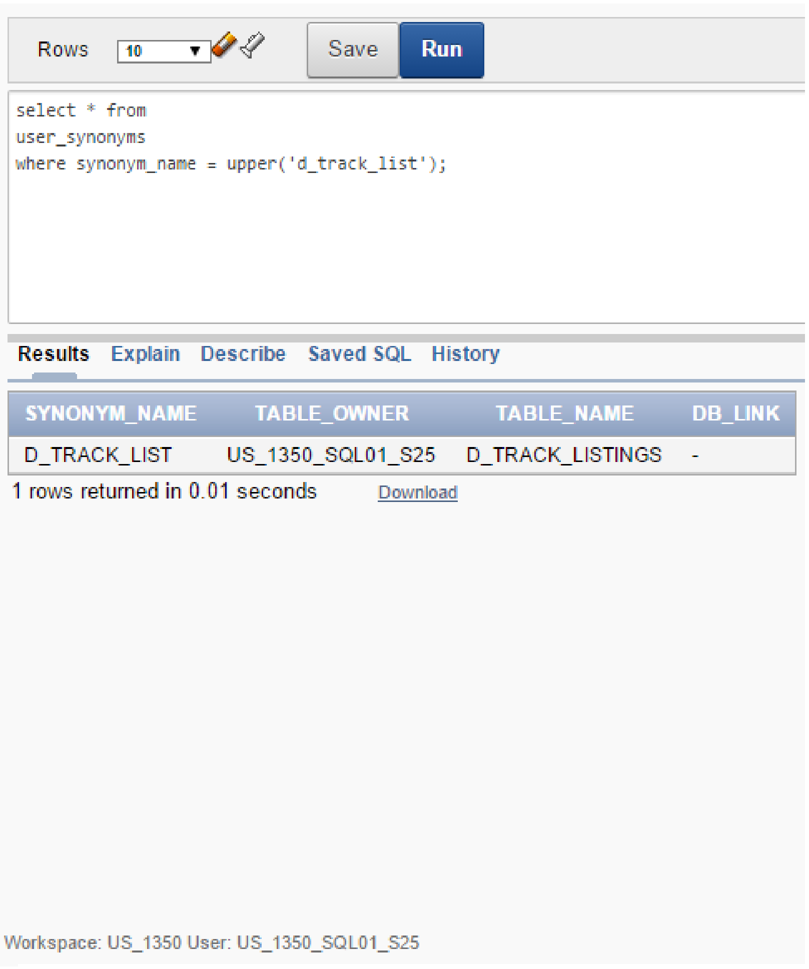
```
create synonym d_track_list
for d_track_listings;
```

Results Explain Describe Saved SQL Histor

```
Synonym created.
```

0.00 seconds

Workspace: US_1350 User: US_1350_SQL01_S25



Rows 10 Save Run

```
select * from
user_synonyms
where synonym_name = upper('d_track_list');
```

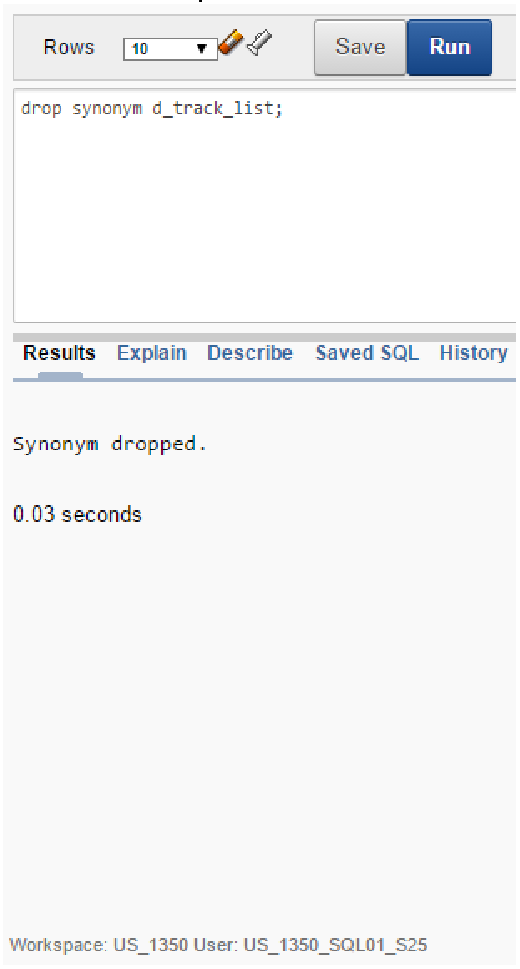
Results Explain Describe Saved SQL History

SYNONYM_NAME	TABLE_OWNER	TABLE_NAME	DB_LINK
D_TRACK_LIST	US_1350_SQL01_S25	D_TRACK_LISTINGS	-

1 rows returned in 0.01 seconds Download

Workspace: US_1350 User: US_1350_SQL01_S25

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



```
drop synonym d_track_list;
```

```
Synonym dropped.
```

0.03 seconds

Workspace: US_1350 User: US_1350_SQL01_S25