



LAB 10: DATABASE INDEX

Prof. Young-Kyoon Suh

Indexes on Oracle (and Other DBMSes)

- Are automatically created without any explicit declaration of creating indexes
 - If you specify primary key or UNIQUE key attribute(s) in a DDL, then Oracle will automatically create the primary index.
- Of course, an index can be created for users' needs.
 - E.g., if you often refer to `EMPLOYEE`'s `FNAME` and/or `LNAME` rather than `SSN`, then you may create a secondary index for `FNAME` and/or `LNAME`.
 - You can explicitly create that index to speed up your queries.

Creating Index on Oracle

- Basic syntax below

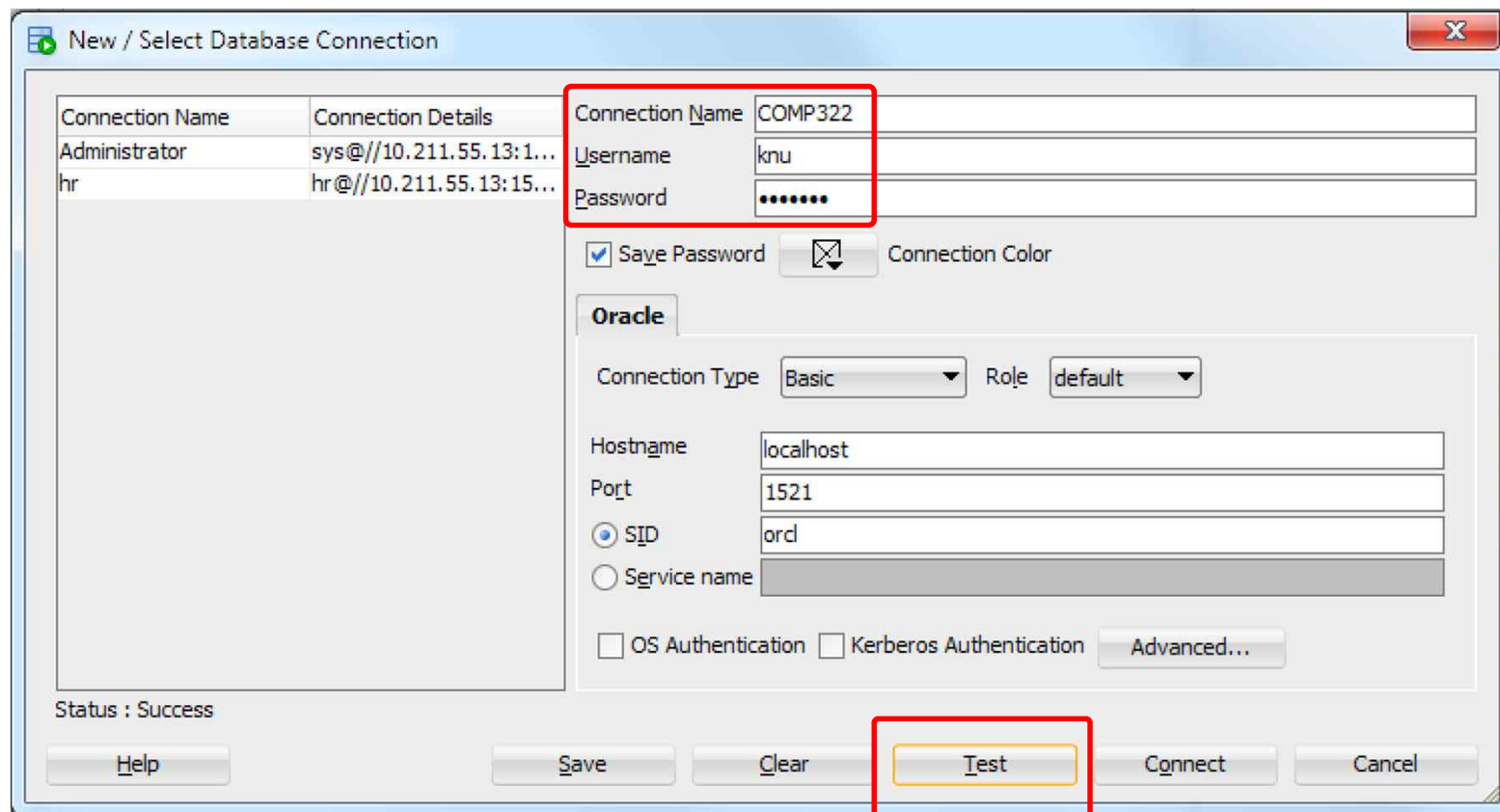
Syntax

```
CRAETE [UNIQUE] INDEX Index_name  
ON Table_name (Column_name (field_name) [ , Column_name, ... ] ) ;
```

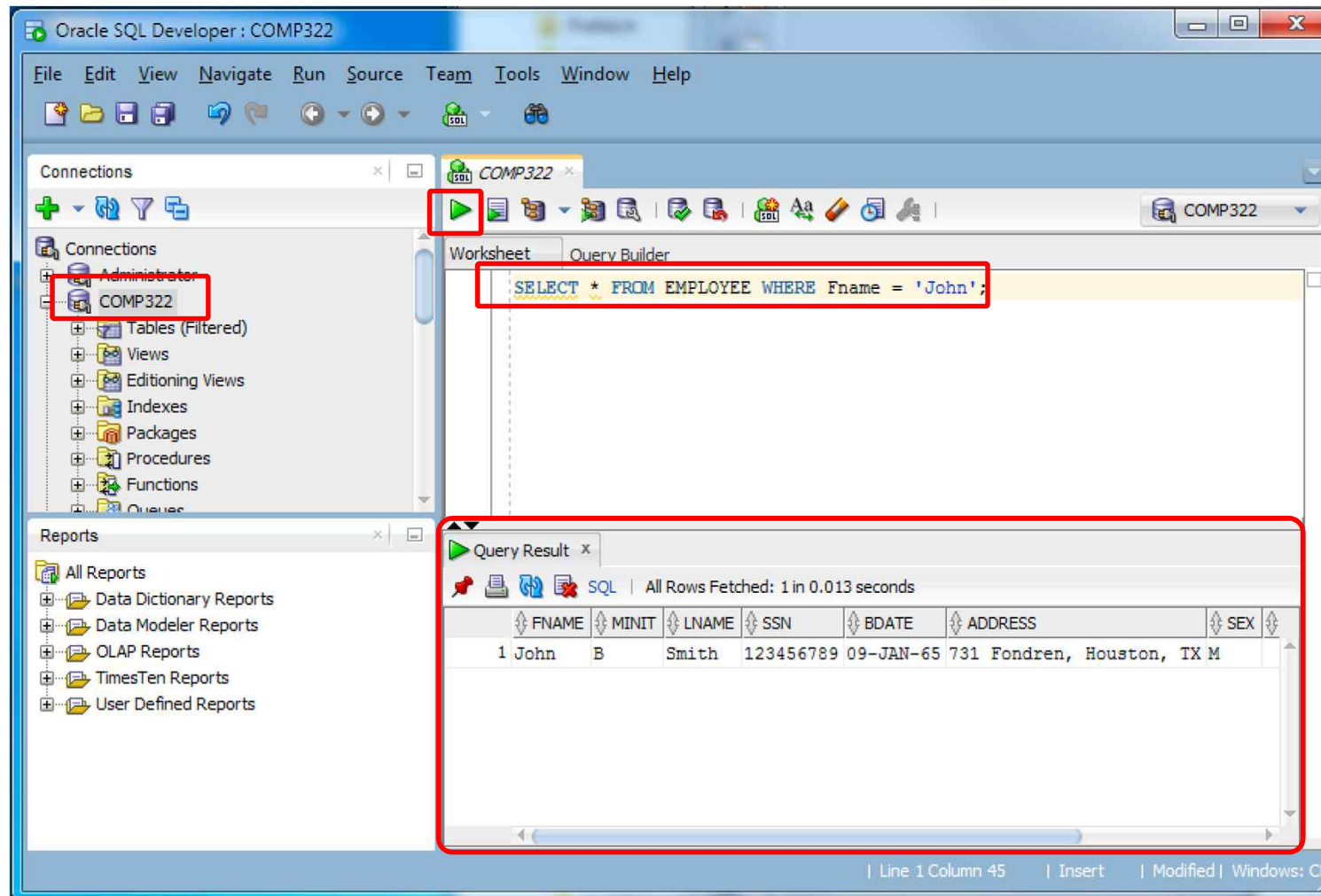
UNIQUE shouldn't be specified for *any* column that may potentially have duplicate values.

Database Connection on SQLDeveloper

- New Connection: named “COMP322”
 - Username/password: knu/comp322



Run a Sample Query



“View Execution Plan” – Table Full Scan

Oracle SQL Developer : COMP322

File Edit View Navigate Run Source Team Tools Window Help

Connections

COMP322

Worksheet

Query Builder

`SELECT * FROM EMPLOYEE WHERE Fname = 'John';`

Query Result x Explain Plan x

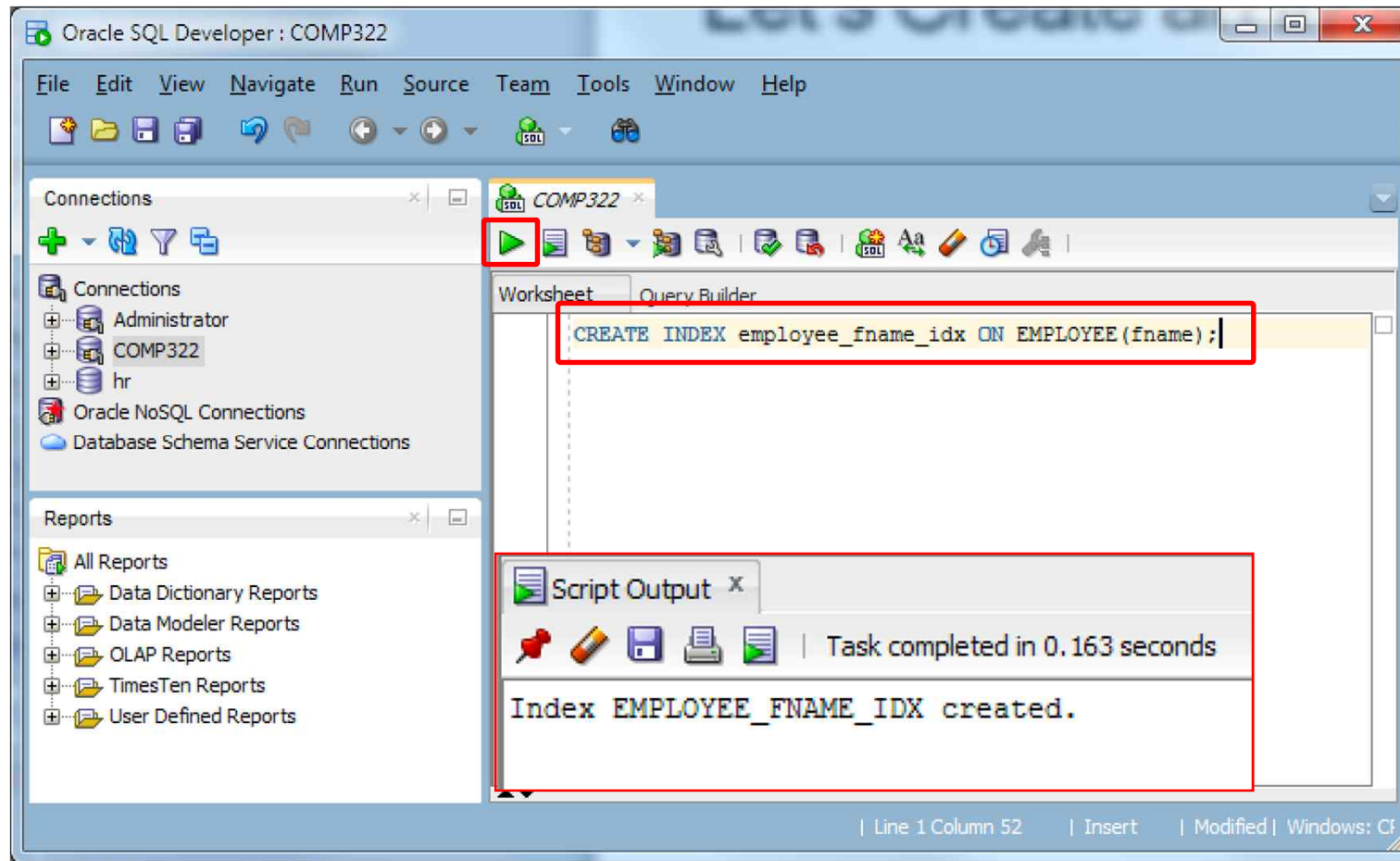
SQL | 0.048 seconds

OPERATION	OBJECT_NAME
SELECT STATEMENT	
TABLE ACCESS (FULL)	EMPLOYEE
Filter Predicates	
FNAME='John'	
Other XML	

- Each EMPLOYEE tuple is scanned for a record containing 'Smith' in its Fname field.

Let's Create an Index for Fname.

- We'll try to speed up query response time.



Let's Create an Index for Fname. (Cont'd)

- Revisit the previous query and view its query execution plan.

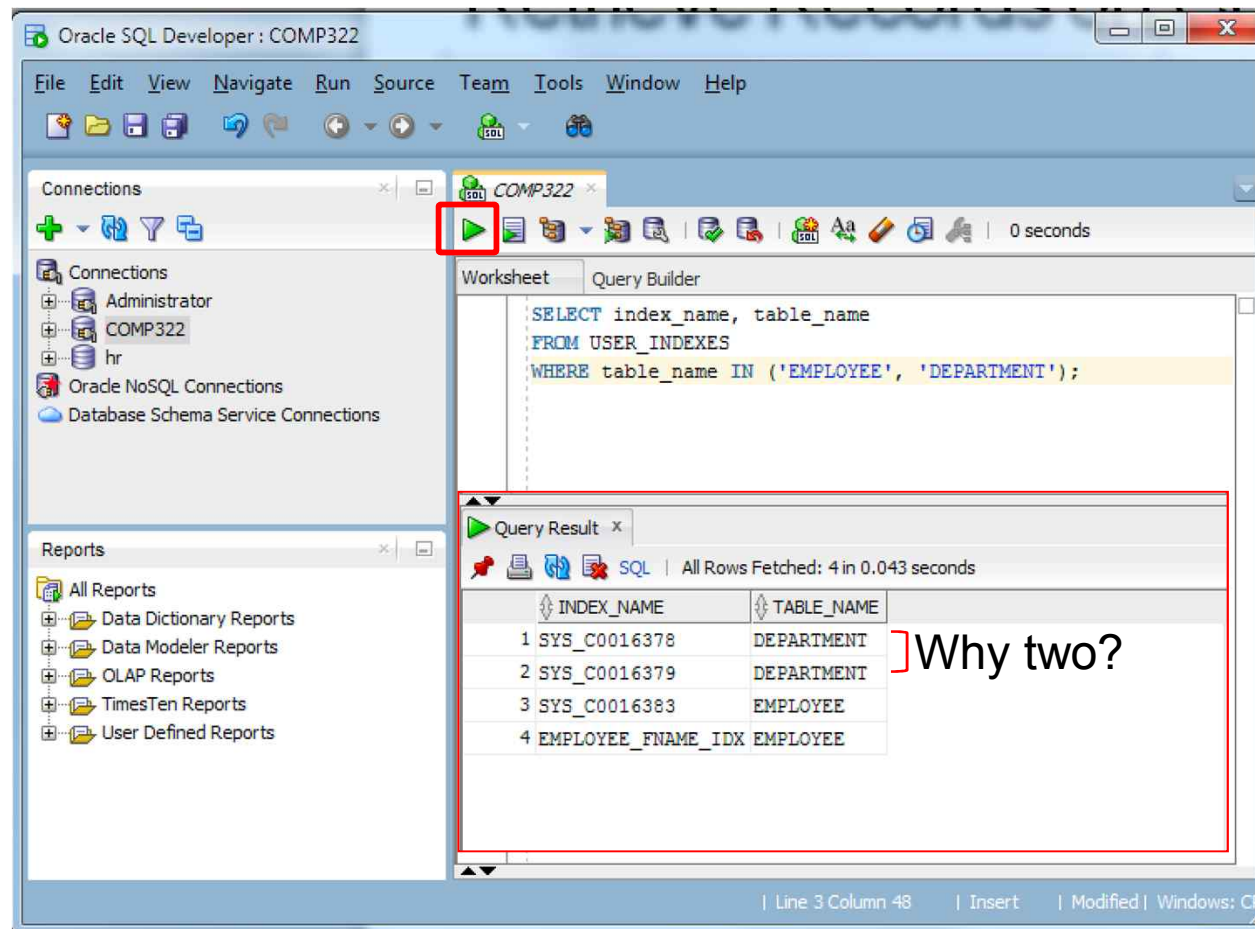
The screenshot shows the Oracle SQL Developer interface. The main window displays the query: `SELECT * FROM EMPLOYEE WHERE Fname = 'John';`. The query execution plan is shown in the bottom right pane, which is titled "Explain Plan". The plan shows the following operations:

OPERATION	OBJECT_NAME
SELECT STATEMENT	
TABLE ACCESS (BY INDEX ROWID BATCHED)	EMPLOYEE
INDEX (RANGE SCAN)	EMPLOYEE_FNAME_IDX
Access Predicates	
FNAME='John'	

The "Access Predicates" section is highlighted with a red box. The status bar at the bottom indicates "Line 1 Column 45 | Insert | Modified | Windows: C".

How to Examine *What Indexes Are Created?*

- Use the USER_INDEXES table.
 - The metadata of the indexes is stored into the table in data dictionary.



The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left lists 'COMP322' as the selected connection. The 'Query Builder' tab is active, displaying the following SQL query:

```
SELECT index_name, table_name
FROM USER_INDEXES
WHERE table_name IN ('EMPLOYEE', 'DEPARTMENT');
```

The 'Query Result' pane at the bottom shows the results of the query. It indicates that 4 rows were fetched in 0.043 seconds. The results are as follows:

	INDEX_NAME	TABLE_NAME
1	SYS_C0016378	DEPARTMENT
2	SYS_C0016379	DEPARTMENT
3	SYS_C0016383	EMPLOYEE
4	EMPLOYEE_FNAME_IDX	EMPLOYEE

A red bracket on the right side of the table groups the two rows for 'DEPARTMENT' and is labeled 'Why two?'. The status bar at the bottom indicates 'Line 3 Column 48 | Insert | Modified | Windows: C'.

How to Examine *What Indexes on Which Columns?*

- Use the USER_IND_COLUMNS table.

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left lists 'Administrator', 'COMP322', 'hr', and 'Oracle NoSQL Connections'. The 'Query Builder' pane in the center contains the following SQL query:

```
SELECT index_name, table_name, column_name
FROM USER_IND_COLUMNS
WHERE table_name IN ('EMPLOYEE', 'DEPARTMENT');
```

The 'Query Result' pane at the bottom displays the results of the query, showing 4 rows of data. The status bar indicates 'All Rows Fetched: 4 in 0.011 seconds'.

	INDEX_NAME	TABLE_NAME	COLUMN_NAME
1	SYS_C0016378	DEPARTMENT	DNUMBER
2	SYS_C0016379	DEPARTMENT	DNAME
3	SYS_C0016383	EMPLOYEE	SSN
4	EMPLOYEE_FNAME_IDX	EMPLOYEE	FNAME

When to Apply Index?

- When indexes are used, we can expect some speedups of query.
- But it's **NOT** always recommended to use the indexes.
- Here are some recommendations for you.

When to Use Index	When Not to Use Index
<ul style="list-style-type: none">- Too many tuples in table- A wide range of column values- Columns referenced in <code>WHERE</code> clause (for selection) or <code>JOIN</code>- 2% ~ 4% of total tuples retrieved by query- Many columns with <code>NULL</code> (<code>NULL</code> values not part of index)	<ul style="list-style-type: none">- Too few tuples in table- Columns seldom referenced in <code>WHERE</code> clause (for selection) or <code>JOIN</code>- 10% ~ 15% of total tuples retrieved by query- Too many updates on table- Index field used with function or <code>NOT</code> operator

When Index is Not Used by Function

The screenshot shows the Oracle SQL Developer interface with the following components:

- Connections:** A list of database connections including Administrator, COMP322, hr, Oracle NoSQL Connections, and Database Schema Service Connections.
- Worksheet:** Contains the SQL query:

```
SELECT *  
FROM EMPLOYEE  
WHERE TO_NUMBER(SSN) = 123456789;
```
- Explain Plan:** Displays the execution plan for the query. The plan shows a **SELECT STATEMENT** operation with a **TABLE ACCESS (FULL)** operation on the **EMPLOYEE** table. The **TABLE ACCESS (FULL)** operation is highlighted with a red box, indicating that a full table scan is performed because the index is not used due to the function on the column.

The status bar at the bottom indicates the current position is Line 3 Column 34, and the window is titled "Saved: COMP322".

When Index is Not Used by NOT Operator

The screenshot shows the Oracle SQL Developer interface with the following components:

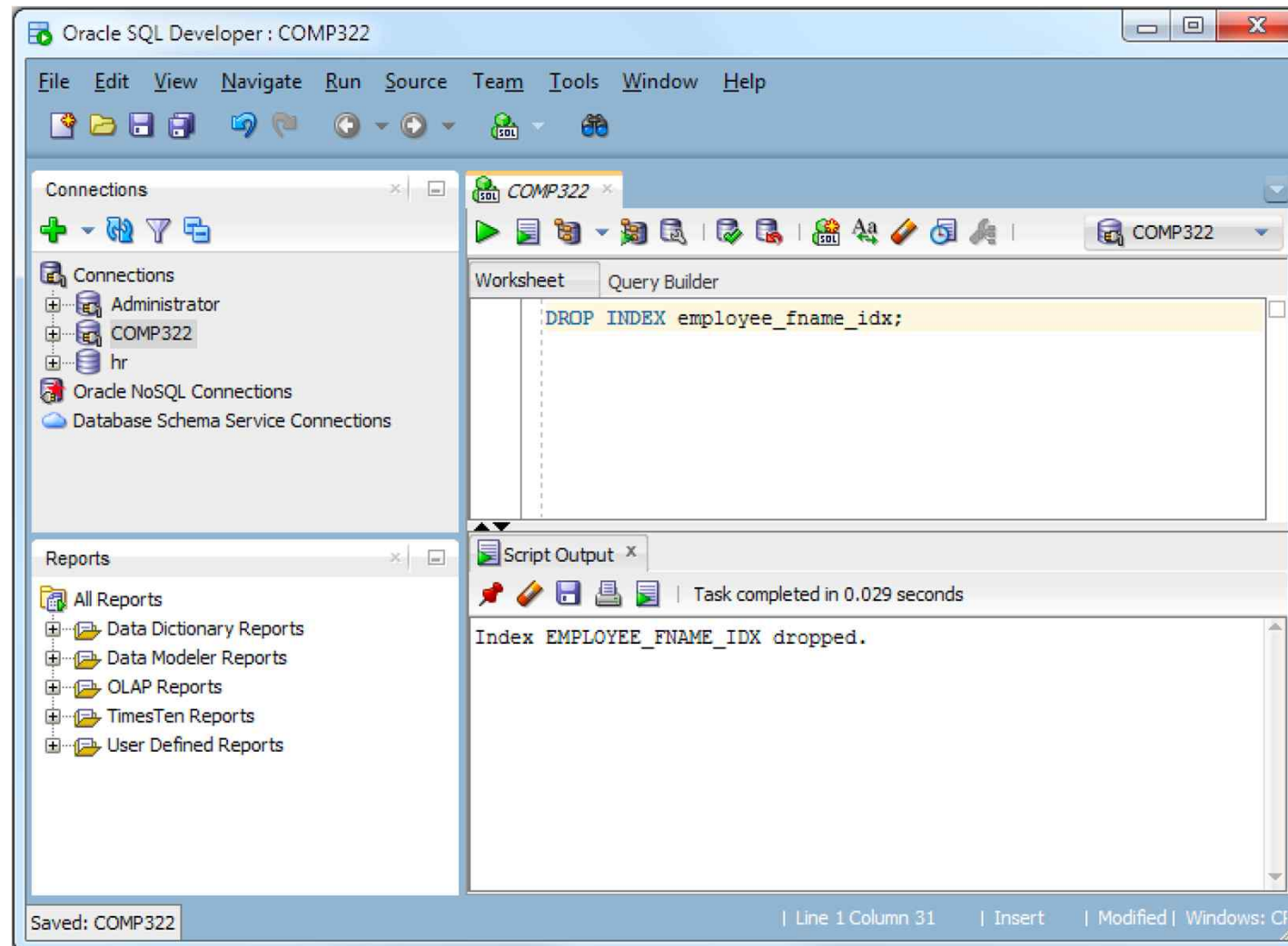
- Connections:** A list of database connections including Administrator, COMP322, hr, Oracle NoSQL Connections, and Database Schema Service Connections.
- Worksheet:** Contains the SQL query:

```
SELECT *  
FROM EMPLOYEE  
WHERE SSN != 1289;
```

The `!=` operator is highlighted with a red box.
- Explain Plan:** Displays the execution plan for the query. The plan shows a **SELECT STATEMENT** operation, which includes a **TABLE ACCESS (FULL)** operation on the **EMPLOYEE** table. The **TABLE ACCESS (FULL)** operation is highlighted with a red box. Below this, the plan shows a **Filter Predicates** operation with the predicate `TO_NUMBER(SSN) <> 1289`.

The status bar at the bottom indicates the file is saved as COMP322, and the cursor is at Line 1 Column 10.

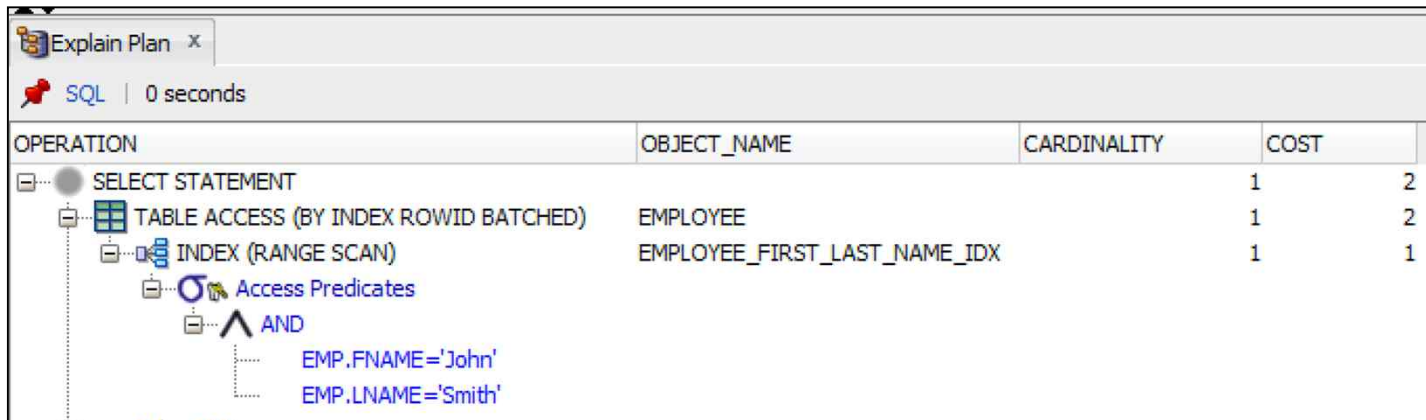
Drop Index: DROP INDEX *index_name*



Lab #10: Playing with Indexes on COMPANY

1) Create an index, named `employee_first_last_name_idx`, on multiple columns of `Fname` and `Lname` on `EMPLOYEE`.

- Show your execution plan using that index for a proper query referencing those two columns (`Fname`, `Lname`).
 - Capture your output as shown below:



OPERATION	OBJECT_NAME	CARDINALITY	COST
SELECT STATEMENT		1	2
TABLE ACCESS (BY INDEX ROWID BATCHED)	EMPLOYEE	1	2
INDEX (RANGE SCAN)	EMPLOYEE_FIRST_LAST_NAME_IDX	1	1
Access Predicates			
AND			
EMP.FNAME='John'			
EMP.LNAME='Smith'			

Lab #10: Playing with Indexes on COMPANY (Cont'd)

2) Create an index, named `dept_dname_mgrssn_idx`, on multiple columns of `Dname` and `Mgr_ssn` on `DEPARTMENT`.

- Write your query of retrieving a record(s) with `Dname = 'Administration'` and `to_number(Mgr_ssn) = 987654321` in `DEPARTMENT`.
- Show your execution plan using that index for the written query. What is your result? Capture your output.

Lab #10: Playing with Indexes on COMPANY (Cont'd)

3) Write your query to view all the indexes created for the PROJECT table.

- What is the index name(s) of the PROJECT table?
- What column(s) is used for creating the indexes?
- Show your output.

Lab #10: Playing with Indexes on COMPANY (Cont'd)

4) How many indexes are created for the WORKS_ON and DEPT_LOCATIONS tables, respectively?

- Tell the number of the indexes for each of the two tables.
- What query did you use? Show your query.
- Show your output of the query.

Lab #10: Playing with Indexes on COMPANY (Cont'd)

5) Create an index for the `Relationship` column in the `DEPENDENT` table.

- Write a proper query to reference that column.
- Capture the output of showing the execution plan of the query and check if your index is used or not.

Lab #10: Playing with Indexes on COMPANY (Cont'd)

[Submission]

- Deadline: **Sunday midnight** (11/18/2018)
- Name your file like:
 - `'lab10-Your_Student_ID.sql'`
 - Put each answer into this `sql` file. If your answer should be written in plain text, then just comment that line.
 - `-- 10-4) # of indexes for WORKS_ON: ..."`
 - Save as image file captured output.
- Zip the `sql` and any screenshot image files.
- Upload the zip file into LMS.

APPENDIX

Configuration of localhost on Your Machine

Step 1: Check your IP address

```
C:\Windows\system32\cmd.exe
C:\Wapache-tomcat-8.5.34\bin>ipconfig

Windows IP Configuration

Ethernet adapter Bluetooth Network Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . : localdomain
    IPv6 Address. . . . . : fdb2:2c26:f4e4:0:7ca8:1c2e:9e4e:e95d
    Temporary IPv6 Address. . . . : fdb2:2c26:f4e4:0:dc75:5abd:fd61:e001
    Link-local IPv6 Address . . . . : fe80::2ca8:1c2e:9e4e:e95d%11
    IPv4 Address. . . . . : 10.211.55.13
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::21c:42ff:fe00:18%11
                               10.211.55.1

Tunnel adapter isatap.{03D27F37-C15E-433D-AE72-F90721FDFD2A}:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 11:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Tunnel adapter isatap.localdomain:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : localdomain

C:\Wapache-tomcat-8.5.34\bin>ping localhost

Pinging YOUNGKYOONS4039 [10.211.55.13] with 32 bytes of data:
Reply from 10.211.55.13: bytes=32 time<1ms TTL=128
Reply from 10.211.55.13: bytes=32 time<1ms TTL=128
```

Step 2: Add an entry like this in
C:\Windows\System32\drivers\etc\hosts.
- Note that you should be able to edit the hosts file. (Run notepad as administrator.)

```
hosts - Notepad
File Edit Format View Help
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host

# localhost name resolution is handled within DNS itself.
127.0.0.1 localhost
10.211.55.13 localhost
# ::1 localhost
```

Step 3: Try ping on localhost.

Step 4: Check the normal response from localhost.