**INDEXES Lab**

**Conor Griffin X00111602**

There are different type of Indexes that are supported by the Oracle 11G DBMS e.g. Primary Key Indexes, unique and non-unique indexes. Let us explore some of them.

**EXERCISE**

When you CREATE or ALTER a table with a PRIMARY KEY and/or a UNIQUE KEY constraints, indexes are automatically created. However, you can manually create the indexes giving the advantage of giving them appropriate names

1. Create the following tables and insert some rows into them. The data should be entered in lowercase. Examine the USER\_INDEXES , USER\_IND\_COLUMNS and USER\_CONSTRAINTS views. How are the constraints named? How are the indexes named? What type of indexes have been created on the mydept and myEmp table? Were you expecting more?

**CREATE TABLE myDept**

**(**

**dept\_id VARCHAR2(9) PRIMARY KEY,**

**dept\_name VARCHAR2(30) NOT NULL UNIQUE**

**);**

**CREATE TABLE myEmp(**

**staff\_id NUMBER(9) PRIMARY KEY,**

**ppsn VARCHAR2(12) NOT NULL UNIQUE,**

**last\_name VARCHAR2(25) NOT NULL,**

**first\_name VARCHAR2(25) NOT NULL,**

**dept\_id VARCHAR2(9),**

**CONSTRAINT fk\_mydept\_dept\_id FOREIGN KEY (dept\_id) REFERENCES mydept(dept\_id)**

**);**

**INSERT INTO myDept VALUES ('1', 'Databases');**

**INSERT INTO myDept VALUES ('2', 'Software Development');**

**INSERT INTO myDept VALUES ('3', 'ADS');**

**INSERT INTO myEmp VALUES(1, '122485C', 'Griffin', 'Conor', '5');**

**INSERT INTO myEmp VALUES(2, '968499H', 'Keogh', 'Ruairi', '3');**

**INSERT INTO myEmp VALUES(3, '874432B', 'Healy', 'Lee', '2');**

**SELECT \* FROM USER\_INDEXES;**

**- Name based on the fact it shows you the index type. All the index types are ‘Normal’ which is the default b-tree index.**

**SELECT \* FROM USER\_IND\_COLUMNS;**

**- Name based on the fact it shows individual columns**

**SELECT \* FROM USER\_CONSTRAINTS;**

**- Name based on the fact it shows the constraint type**

**EXERCISE**

1. You should notice that an index for the Foreign Key constraint does not exist. If there are lots of queries like

**SELECT last\_name, first\_name, dept\_name**

**FROM mydept md, myemp me  
WHERE md.dept\_id=me.dept\_id**

you will end up with full table scans. Create an appropriate index for the foreign key using the CREATE INDEX command. Check USER\_INDEXES view to see they index definition

**CREATE INDEX fk\_index ON myEmp (dept\_id);**

1. Drop myEmp and myDept tables. Check the indexes data dictionary view to see do the indexes exist.

**DROP TABLE myEmp;**

**DROP TABLE myDept;**

**SELECT \* FROM USER\_INDEXES;**

**The indexes that we could see before from the data dictionary are now gone.**

1. You can explicitly name a PRIMARY KEY index whilst creating a table. Review the syntax and re-create the myDept table as follows. Then query USER\_CONSTRAINTS, USER\_IND\_COLUMNS and USER\_INDEXES for this table. What do you notice?

**CREATE TABLE myDept**

**(**

**dept\_id VARCHAR2(9) PRIMARY KEY USING**

**INDEX (CREATE UNIQUE INDEX dept\_id\_pk\_idx**

**ON mydept(dept\_id)),**

**dept\_name VARCHAR2(30) NOT NULL UNIQUE USING**

**INDEX (CREATE UNIQUE INDEX dept\_name\_idx**

**ON mydept(dept\_name))**

**);**

**After querying user\_constraints, user\_ind\_columns and user\_indexes I noticed that the indexes created where manually created with the table instead of individually.**

1. Now recreate table myEmp. Make sure to explicitly name the indexes for the primary key AND the unique key. **Note** add the Foreign Key Index after you have created the table. Check the USER\_CONSTRAINTS, USER\_IND\_COLUMNS and USER\_INDEXES for this table to see what has been created in the data dictionary.

**CREATE TABLE myEmp(**

**staff\_id NUMBER(9) PRIMARY KEY USING**

**INDEX (CREATE UNIQUE INDEX staff\_id\_pk\_idx**

**ON myemp(staff\_id)),**

**ppsn VARCHAR2(12) NOT NULL UNIQUE USING**

**INDEX (CREATE UNIQUE INDEX ppsn\_idx**

**ON myemp(ppsn)),**

**last\_name VARCHAR2(25) NOT NULL,**

**first\_name VARCHAR2(25) NOT NULL,**

**dept\_id VARCHAR2(9),**

**CONSTRAINT fk\_mydept\_dept\_id FOREIGN KEY (dept\_id) REFERENCES mydept(dept\_id)**

**);**

1. Execute the following SQL statements individually. Explain the reason for the error.

**CREATE TABLE test (col1 NUMBER);**

**CREATE INDEX test\_col1\_i ON test(col1);**

**DROP TABLE test;**

**DROP INDEX test\_col1\_i;**

**The Index is already dropped when you dropped the table before-hand hence why you get the error ‘specified index does not exist’**

1. Now execute these statements individually. What do you concur?

**CREATE TABLE test (col1 NUMBER);**

**CREATE INDEX test\_col1\_i ON test(col1);**

**DROP INDEX test\_col1\_i;**

**DROP TABLE test;**

**The index was dropped before the table was dropped which is allowed, so there was no errors generated as it is allowed to be dropped in this order.**

**EXERCISE**:

Review the ERD and accompanying notes below. Accordingly, create the following tables:

1. **ittcourse** that has the following columns course\_id (primary key) , course\_name, course\_description. course \_name is mandatory.

**CREATE TABLE ittcourse (**

**course\_id NUMBER(9) PRIMARY KEY USING**

**INDEX (CREATE UNIQUE INDEX ittcourse\_id\_pk\_idx**

**ON ittcourse(course\_id)),**

**course\_name VARCHAR2(40) NOT NULL,**

**course\_description VARCHAR2(40)**

**);**

1. **ittstudent** that has columns stud\_id (primary key),ppsn (unique key), firstname, lastname, phone. All columns are mandatory.

**CREATE TABLE ittstudent (**

**stud\_id NUMBER(9) PRIMARY KEY USING**

**INDEX (CREATE UNIQUE INDEX student\_id\_pk\_idx**

**ON ittstudent(stud\_id)),**

**ppsn VARCHAR2(12) NOT NULL UNIQUE USING**

**INDEX (CREATE UNIQUE INDEX ppsn\_idx**

**ON ittstudent(ppsn)),**

**first\_name VARCHAR2(25) NOT NULL,**

**last\_name VARCHAR2(25) NOT NULL,**

**phone VARCHAR2(15) NOT NULL**

**);**

1. **ittstudentCourse** table is an table containing course\_id and stud\_id (as a composite primary key) and enrolment\_date, completion\_date

**CREATE TABLE ittstudentcourse (**

**course\_id NUMBER(9),**

**stud\_id NUMBER(9),**

**enrolment\_date VARCHAR2(20),**

**completion\_date VARCHAR2(20),**

**CONSTRAINT pk\_ittstudentcourse\_con PRIMARY KEY (course\_id, stud\_id)**

**);**

ITTCourse

ITTStudentCourse

ITTStudent

**Notes**

Indexes must be explicitly named as follows:

* The primary key **index** for ittcourse must be called ittcourse\_id\_pk\_idx
* The primary key **index** for ittstudent must be called student\_id\_pk\_idx
* The primary key **index** for ittstudentcourse must be called ittstudcourse\_ids\_comppk\_idx and the primary key constraint must be called pk\_ittstudentcourse\_con. **Note** you need to create the index when you are creating the Table constraint
* After creating the ITTStudentsCourse table add indexes for the 2 foreign key constraints

**CREATE INDEX fk\_course\_id\_index ON ittstudentcourse (course\_id);**

**CREATE INDEX fk\_stud\_id\_index ON ittstudentcourse (stud\_id);**

**Exercise**

1. Create a nonunique index on course\_name on the ittcourse table. Then DROP the index!

**CREATE INDEX course\_name\_idx ON ittcourse(course\_name);**

**DROP INDEX course\_name\_idx;**

1. Insert 5 rows each into course and student tables respectively and “register” a number of students on the courses.

**INSERT INTO ittcourse VALUES (1, 'Law' , 'Law & Order');**

**INSERT INTO ittcourse VALUES (2, 'History', 'The Secret To History');**

**INSERT INTO ittcourse VALUES (3, 'Computer Science', 'Computing');**

**INSERT INTO ittcourse VALUES (4, 'Creative Digital Media', 'Digital Media');**

**INSERT INTO ittcourse VALUES (5, 'Business', 'Rule the world');**

**INSERT INTO ittstudent VALUES (1, '34226Y', 'Conor', 'Griffin', '0853345678');**

**INSERT INTO ittstudent VALUES (2, '88712T', 'Damien', 'Ward', '0838798877');**

**INSERT INTO ittstudent VALUES (3, '23455W', 'Graham', 'Lalor', '0879822123');**

**INSERT INTO ittstudent VALUES (4, '88997S', 'Lee', 'Healy', '0868765567');**

**INSERT INTO ittstudent VALUES (5, '90554R', 'Lucy', 'Bell', '0834434654');**

1. Let us execute a few queries and see what index was used by examining the Execution plan as follows (**note** change the values to suit your data):

**set autotrace on explain**

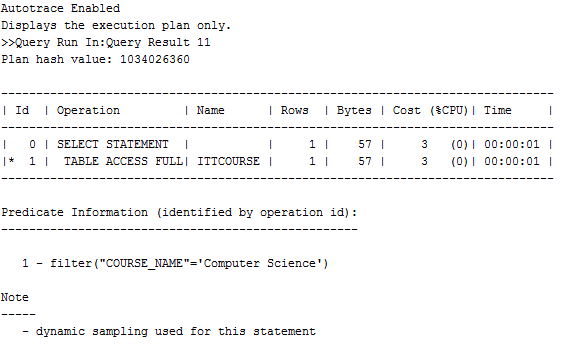
**select \* from ittcourse**

**where course\_name='networking';**

**set autotrace on explain**

**select \* from ittcourse**

**where course\_name='Computer Science';**



The execution Plan can be examined under “script output” tab beside the query result. You should have noticed for the above execution plan that NO index was used. This is expected as we have not created an index on course\_name. The ittcourse table was scanned using the predicate filter( "COURSE\_NAME"='networking')

Examine the Execution Plans for the queries below.Capture the Execution Plan Output. Interpret it in the context of whether an index was used or not; how the index was accessed; how the table was accessed.

**set autotrace on explain**

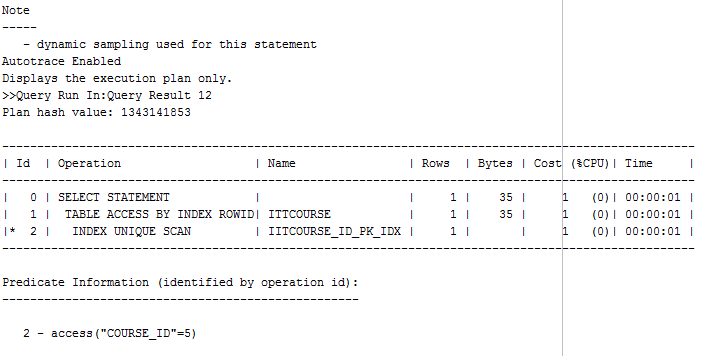
**select course\_name from ittcourse**

**where course\_id = 999;**

**set autotrace on explain**

**select course\_name from ittcourse**

**where course\_id = 5;**



**set autotrace on explain**

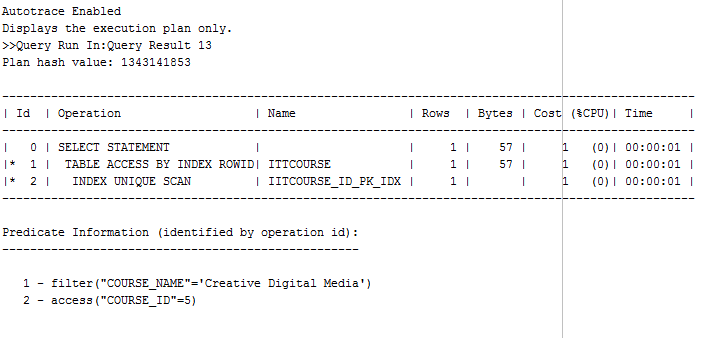
**select \* from ittcourse**

**where course\_id = 999 and course\_name='databases';**

**set autotrace on explain**

**select \* from ittcourse**

**where course\_id = 5 and course\_name='Creative Digital Media';**



**Index was used as it uses course\_id = 5 as an index rowed to access**

**[note to turn off autotrace SET AUTOTRACE OFF;]**

**set autotrace off;**

1. Examine the execution plan for the query below , identifying whether an index was used of not. What was the name of the index if an index was used? If no Index was used, create an appropriate one with a CREATE INDEX stu\_last\_firstname\_indx. Rerun your query what do you notice?

**set autotrace on explain**

**SELECT firstname,lastname**

**FROM ittstudent**

**WHERE lastname='smith'**

**AND firstname='james';**

**set autotrace on explain**

**SELECT first\_name, last\_name**

**FROM ittstudent**

**WHERE last\_name='Griffin'**

**AND first\_name='Conor';**

**NO INDEX WAS USED IN THIS QUERY**

**CREATE INDEX stu\_last\_firstname\_indx ON ittstudent(last\_name, first\_name);**

1. Run the query below. Examine the execution plan. The Cost Based Optimizer CBO execution plan can be overridden by using a hint **/\*+ index(i stu\_last\_firstname\_indx) \*/ .** We are forcing the Cost Based Optimizer ( CBO) to use the index.

Note: change the command to use the index you created.

**set autotrace on explain**

**SELECT /\*+ index(ittstudent stu\_last\_firstname\_indx) \*/ firstname,lastname,ppsn**

**FROM ittstudent**

**WHERE lastname='smith';**

**Remove the index you created in 14).**

**set autotrace on explain**

**SELECT /\*+ index(ittstudent stu\_last\_firstname\_indx)\*/ first\_name,last\_name,ppsn**

**FROM ittstudent**

**WHERE last\_name='Healy';**

