**Session 12 Adding /Viewing TABLE CONSTRAINTS**

Last login: Tue Mar 22 11:40:16 2022 from 10.30.2.33

[student@oracledb19c ~]$ **su - oracle**

Password:

Last login: Tue Mar 22 11:40:24 EDT 2022 on pts/0

The Oracle base remains unchanged with value /opt/oracle/app/oracle

[oracle@oracledb19c ~]$ **ls -l**

total 44

-rw-r--r-- 1 oracle dba 0 Jan 28 14:36 conn

-rw-r--r-- 1 oracle dba 0 Jan 28 14:32 exit

-rw-r--r-- 1 oracle dba 396 Jan 29 12:53 profile.txt

-rw-r--r-- 1 oracle dba 9806 Feb 20 10:31 **scott2.sql**

-rw-r--r-- 1 oracle dba 22542 Feb 20 10:14 scott.sql

-rw-r--r-- 1 oracle dba 0 Jan 28 14:36 SELECT

-rw-r--r-- 1 oracle dba 47 Jan 6 11:48 setport.sql

**\* If you did NOT create user SCOTT, then you need to run scott2.sql script (as SYSDBA) and create 7 tables in Scott’s schema \***

[oracle@oracledb19c ~]$ **sqlplus /nolog**

SQL\*Plus: Release 19.0.0.0.0 - Production on Fri Mar 25 10:06:18 2022

Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

SQL> **set pagesize 120**

SQL> **conn scott/tiger**

Connected.

SQL> **select tname from tab;**

TNAME

----------------------------------------------------------------------

EMP

DEPT

BONUS

SALGRADE

DUMMY

CUSTOMER

ORD

7 rows selected.

SQL> **desc customer**

Name Null? Type

----------------------------------------- -------- ------------------

CUSTID NOT NULL NUMBER(6)

NAME CHAR(45)

ADDRESS CHAR(40)

CITY CHAR(30)

STATE CHAR(2)

ZIP CHAR(9)

AREA NUMBER(3)

PHONE CHAR(9)

REPID NOT NULL NUMBER(4)

CREDITLIMIT NUMBER(9,2)

**COMMENTS LONG**

SQL> **conn / as sysdba**

Connected.

**\* Let’s create a new Tablespace INDX just for the index storage \***

SQL> **CREATE TABLESPACE indx**

**DATAFILE '/opt/oracle/app/oracle/oradata/DISK2/indx01.dbf' SIZE 10M;**

Tablespace created.

SQL> **ALTER USER tom QUOTA 3M ON indx;** 🡪 **Tom can put his Indexes here**

User altered.

**\* Let’s add 3 crucial Scott’s tables into TOM’s schema \***

SQL> **CREATE TABLE tom.dept AS SELECT \* FROM scott.dept;**

Table created.

SQL> **CREATE TABLE tom.emp AS SELECT \* FROM scott.emp;**

Table created.

SQL> CREATE TABLE tom.customer AS SELECT \* FROM scott.customer;

CREATE TABLE tom.customer AS SELECT \* FROM scott.customer

\*

ERROR at line 1:

ORA-00997: illegal use of LONG datatype

**\* Let’s drop column Comments of LONG datatype from this table \***

SQL> **ALTER TABLE scott.customer DROP COLUMN comments;**

Table altered.

SQL> **CREATE TABLE tom.customer AS SELECT \* FROM scott.customer;**

Table created.

SQL> **conn tom/cat**

Connected.

SQL> **SELECT \* FROM dept;**

DEPTNO DNAME LOC

---------- -------------- -------------

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

SQL> **SELECT \* FROM emp;**

EMPNO ENAME JOB MGR HIREDATE SAL

---------- ---------- --------- ---------- --------- ---------- ------

COMM DEPTNO

-------------

7839 KING PRESIDENT 17-NOV-81 5000

10

7698 BLAKE MANAGER 7839 01-MAY-81 2850

30

7782 CLARK MANAGER 7839 09-JUN-81 2450

10

7566 JONES MANAGER 7839 02-APR-81 2975

20

7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30

7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30

7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30

7900 JAMES CLERK 7698 03-DEC-81 950

30

7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30

7902 FORD ANALYST 7566 03-DEC-81 3000

20

7369 SMITH CLERK 7902 17-DEC-80 800

20

7788 SCOTT ANALYST 7566 09-DEC-82 3000

20

7876 ADAMS CLERK 7788 12-JAN-83 1100

20

7934 MILLER CLERK 7782 23-JAN-82 1300

10

14 rows selected.

SQL> **SELECT \* FROM customer;**

CUSTID NAME

---------- ---------------------------------------------

ADDRESS CITY ST

---------------------------------------- -----------------------------

ZIP AREA PHONE REPID CREDITLIMIT

--------- ---------- --------- ---------- -----------

100 JOCKSPORTS

345 VIEWRIDGE BELMONT CA

96711 415 598-6609 7844 5000

101 TKB SPORT SHOP

490 BOLI RD. REDWOOD CITY CA

94061 415 368-1223 7521 10000

102 VOLLYRITE

9722 HAMILTON BURLINGAME CA

95133 415 644-3341 7654 7000

103 JUST TENNIS

HILLVIEW MALL BURLINGAME CA

97544 415 677-9312 7521 3000

104 EVERY MOUNTAIN

574 SURRY RD. CUPERTINO CA

93301 408 996-2323 7499 10000

105 K + T SPORTS

3476 EL PASEO SANTA CLARA CA

91003 408 376-9966 7844 5000

106 SHAPE UP

908 SEQUOIA PALO ALTO CA

94301 415 364-9777 7521 6000

107 WOMENS SPORTS

VALCO VILLAGE SUNNYVALE CA

93301 408 967-4398 7499 10000

108 NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER

98 LONE PINE WAY HIBBING MN

55649 612 566-9123 7844 8000

9 rows selected.

SQL> **desc emp**

Name Null? Type

----------------------------------------- -------- ------------------

EMPNO NUMBER(6)

ENAME CHAR(10)

JOB CHAR(9)

MGR NUMBER(4)

HIREDATE DATE

SAL NUMBER(7,2)

COMM NUMBER(7,2)

DEPTNO NOT NULL NUMBER(2)

SQL> **desc customer**

Name Null? Type

----------------------------------------- -------- ------------------

CUSTID NUMBER(6)

NAME CHAR(45)

ADDRESS CHAR(40)

CITY CHAR(30)

STATE CHAR(2)

ZIP CHAR(9)

AREA NUMBER(3)

PHONE CHAR(9)

REPID NOT NULL NUMBER(4)

CREDITLIMIT NUMBER(9,2)

SQL> **ALTER TABLE dept**

**ADD (CONSTRAINT dept\_deptno\_pk PRIMARY KEY (deptno)**

**DEFERRABLE using index TABLESPACE indx,**

**CONSTRAINT dept\_dname\_uk UNIQUE(dname) DISABLE);**

**\* PK constraint here is of IMMEDIATE subtype, it behaves like non-deferrable one (checks each Statement separately), but in the future may be switched to DEFERRED subtype (checks all Statements at Commit time). UK constraint is turned off (NO index was created) and will not check for violations at all. It may be enabled later, if there are no inconsistencies (duplicate names). \***

Table altered.

SQL> **ALTER TABLE emp**

**ADD (CONSTRAINT emp\_empno\_pk PRIMARY KEY (empno)**

**using index TABLESPACE indx,**

**CONSTRAINT emp\_deptno\_fk FOREIGN KEY (deptno) REFERENCES dept (deptno)**

**DEFERRABLE INITIALLY IMMEDIATE,**

**CONSTRAINT emp\_sal\_ck CHECK (sal BETWEEN 500 AND 4000)**

**ENABLE NOVALIDATE);**

**\* FK constraint here is of IMMEDIATE subtype as well, it behaves like non-deferrable one (checks each Statement separately), but in the future may be switched to DEFERRED subtype (checks all Statements at Commit time). CK constraint is turned on (Unique Index was created by server) and will check only INCOMING violations, but for now it will not inspect existing Salary values. It may be enabled later, if there are no inconsistencies (Salary out of range). \***

Table altered.

SQL> **ALTER TABLE customer**

**ADD (CONSTRAINT cust\_custid\_pk PRIMARY KEY (custid)**

**using index TABLESPACE indx,**

**CONSTRAINT cust\_repid\_fk FOREIGN KEY (repid) REFERENCES emp (empno)**

**DEFERRABLE INITIALLY DEFERRED,**

**CONSTRAINT cust\_credlim\_ck CHECK (creditlimit < 10000)**

**ENABLE NOVALIDATE);**

**\* FK constraint here is of DEFERRED subtype, it behaves like deferred one (checks all Statements at Commit time), but later may be switched to the Immediate subtype.**

**CK constraint is turned on (Unique Index was created by server) and will check only INCOMING violations, but for now it will not inspect existing Limit values. It may be enabled later, if there are no inconsistencies (Limit out of range). \***

Table altered.

SQL> **DESC dba\_constraints**

ERROR:

ORA-04043: object "SYS"."DBA\_CONSTRAINTS" does not exist

SQL> **set role select\_catalog\_role;**

**\* Do not forget, that TOM was granted this role that enables browsing of Data Dictionary, but this role was not made a DEFAULT one. It needs to be enabled manually by DBA or TOM himself. \***

Role set.

SQL> **DESC dba\_constraints**

Name Null? Type

----------------------------------------- -------- ------------------

OWNER VARCHAR2(128)

CONSTRAINT\_NAME NOT NULL VARCHAR2(128)

CONSTRAINT\_TYPE VARCHAR2(1)

TABLE\_NAME NOT NULL VARCHAR2(128)

SEARCH\_CONDITION LONG

SEARCH\_CONDITION\_VC VARCHAR2(4000)

R\_OWNER VARCHAR2(128)

R\_CONSTRAINT\_NAME VARCHAR2(128)

DELETE\_RULE VARCHAR2(9)

STATUS VARCHAR2(8)

DEFERRABLE VARCHAR2(14)

DEFERRED VARCHAR2(9)

VALIDATED VARCHAR2(13)

GENERATED VARCHAR2(14)

BAD VARCHAR2(3)

RELY VARCHAR2(4)

LAST\_CHANGE DATE

INDEX\_OWNER VARCHAR2(128)

INDEX\_NAME VARCHAR2(128)

INVALID VARCHAR2(7)

VIEW\_RELATED VARCHAR2(14)

ORIGIN\_CON\_ID NUMBER

SQL> **SELECT constraint\_name, constraint\_type, search\_condition,**

**status, deferrable, deferred, validated, table\_name**

**FROM dba\_constraints**

**WHERE table\_name IN ('DEPT','EMP','CUSTOMER')**

**AND owner = 'TOM'**

**ORDER BY 8, 2, 1;**

CONSTRAINT\_NAME C

--------------------------------------

SEARCH\_CONDITION

----------------------------------------------------------------------

STATUS DEFERRABLE DEFERRED VALIDATED

-------- -------------- --------- -------------

TABLE\_NAME

----------------------------------------------------------------------

CUST\_CREDLIM\_CK C

creditlimit < 10000

ENABLED NOT DEFERRABLE IMMEDIATE NOT VALIDATED

CUSTOMER

SYS\_C008302 C

"REPID" IS NOT NULL

ENABLED NOT DEFERRABLE IMMEDIATE VALIDATED

CUSTOMER

CUST\_CUSTID\_PK P

ENABLED NOT DEFERRABLE IMMEDIATE VALIDATED

CUSTOMER

CUST\_REPID\_FK R

ENABLED DEFERRABLE DEFERRED VALIDATED

CUSTOMER

DEPT\_DEPTNO\_PK P

ENABLED DEFERRABLE IMMEDIATE VALIDATED

DEPT

DEPT\_DNAME\_UK U

DISABLED NOT DEFERRABLE IMMEDIATE NOT VALIDATED

DEPT

EMP\_SAL\_CK C

sal BETWEEN 500 AND 4000

ENABLED NOT DEFERRABLE IMMEDIATE NOT VALIDATED

EMP

SYS\_C008301 C

"DEPTNO" IS NOT NULL

ENABLED NOT DEFERRABLE IMMEDIATE VALIDATED

EMP

EMP\_EMPNO\_PK P

ENABLED NOT DEFERRABLE IMMEDIATE VALIDATED

EMP

EMP\_DEPTNO\_FK R

ENABLED DEFERRABLE IMMEDIATE VALIDATED

EMP

10 rows selected.

**\* In SQL you can see Constraint parameters for more than one table and also Check (Search)Condition for all CHECK constraints when browsing dba\_constraints view, but this view does not show you the column name(s) \***

**\* If you want to see the Column Name(s) for each constraint you go for dba\_cons\_columns view \***

SQL> **ALTER TABLE emp**

**ADD CONSTRAINT emp\_job\_hdate\_uk UNIQUE(job,hiredate);**

Table altered.

SQL> **DESC dba\_cons\_columns**

Name Null? Type

----------------------------------------- -------- ------------------

OWNER NOT NULL VARCHAR2(128)

CONSTRAINT\_NAME NOT NULL VARCHAR2(128)

TABLE\_NAME NOT NULL VARCHAR2(128)

COLUMN\_NAME VARCHAR2(4000)

POSITION NUMBER

SQL> **SELECT constraint\_name, column\_name, position**

**FROM dba\_cons\_columns**

**WHERE owner = 'TOM'**

**AND table\_name = 'EMP'**

**AND owner = 'TOM'**

**ORDER BY 4,2;**

CONSTRAINT\_NAME

----------------------------------------------------------------------

COLUMN\_NAME POSITION

----------------------------------------------------------------------

EMP\_DEPTNO\_FK

DEPTNO 1

EMP\_EMPNO\_PK

EMPNO 1

EMP\_JOB\_HDATE\_UK

JOB 1

EMP\_JOB\_HDATE\_UK

HIREDATE 2

EMP\_SAL\_CK

SAL

SYS\_C008301

DEPTNO

6 rows selected.

**\* Check constraints do not have value in a Position column. This column is used to specify whether is constraint SINGLE --> 1 or the COMPOSITE one --> 1 and 2(and 3 etc.) for the SAME Constraint Name \***

SQL> **DESC dba\_indexes**

Name Null? Type

----------------------------------------- -------- ------------------

OWNER NOT NULL VARCHAR2(128)

INDEX\_NAME NOT NULL VARCHAR2(128)

INDEX\_TYPE VARCHAR2(27)

TABLE\_OWNER NOT NULL VARCHAR2(128)

TABLE\_NAME NOT NULL VARCHAR2(128)

TABLE\_TYPE VARCHAR2(11)

UNIQUENESS VARCHAR2(9)

COMPRESSION VARCHAR2(13)

PREFIX\_LENGTH NUMBER

TABLESPACE\_NAME VARCHAR2(30)

INI\_TRANS NUMBER

MAX\_TRANS NUMBER

INITIAL\_EXTENT NUMBER

NEXT\_EXTENT NUMBER

MIN\_EXTENTS NUMBER

MAX\_EXTENTS NUMBER

PCT\_INCREASE NUMBER

PCT\_THRESHOLD NUMBER

INCLUDE\_COLUMN NUMBER

FREELISTS NUMBER

FREELIST\_GROUPS NUMBER

PCT\_FREE NUMBER

LOGGING VARCHAR2(3)

BLEVEL NUMBER

LEAF\_BLOCKS NUMBER

DISTINCT\_KEYS NUMBER

AVG\_LEAF\_BLOCKS\_PER\_KEY NUMBER

AVG\_DATA\_BLOCKS\_PER\_KEY NUMBER

CLUSTERING\_FACTOR NUMBER

STATUS VARCHAR2(8)

NUM\_ROWS NUMBER

SAMPLE\_SIZE NUMBER

LAST\_ANALYZED DATE

DEGREE VARCHAR2(40)

INSTANCES VARCHAR2(40)

PARTITIONED VARCHAR2(3)

TEMPORARY VARCHAR2(1)

GENERATED VARCHAR2(1)

SECONDARY VARCHAR2(1)

BUFFER\_POOL VARCHAR2(7)

FLASH\_CACHE VARCHAR2(7)

CELL\_FLASH\_CACHE VARCHAR2(7)

USER\_STATS VARCHAR2(3)

DURATION VARCHAR2(15)

PCT\_DIRECT\_ACCESS NUMBER

ITYP\_OWNER VARCHAR2(128)

ITYP\_NAME VARCHAR2(128)

PARAMETERS VARCHAR2(1000)

GLOBAL\_STATS VARCHAR2(3)

DOMIDX\_STATUS VARCHAR2(12)

DOMIDX\_OPSTATUS VARCHAR2(6)

FUNCIDX\_STATUS VARCHAR2(8)

JOIN\_INDEX VARCHAR2(3)

IOT\_REDUNDANT\_PKEY\_ELIM VARCHAR2(3)

DROPPED VARCHAR2(3)

VISIBILITY VARCHAR2(9)

DOMIDX\_MANAGEMENT VARCHAR2(14)

SEGMENT\_CREATED VARCHAR2(3)

ORPHANED\_ENTRIES VARCHAR2(3)

INDEXING VARCHAR2(7)

AUTO VARCHAR2(3)

CONSTRAINT\_INDEX VARCHAR2(3)

SQL> **SELECT index\_name, index\_type, uniqueness**

**FROM dba\_indexes**

**WHERE index\_name IN (SELECT constraint\_name**

**FROM dba\_constraints**

**WHERE owner = 'TOM'**

**AND table\_name IN ('DEPT','EMP','CUSTOMER') )**

**ORDER BY 1;**

INDEX\_NAME

----------------------------------------------------------------------

INDEX\_TYPE UNIQUENES

--------------------------- ---------

CUST\_CUSTID\_PK

NORMAL UNIQUE

DEPT\_DEPTNO\_PK

NORMAL NONUNIQUE

EMP\_EMPNO\_PK

NORMAL UNIQUE

EMP\_JOB\_HDATE\_UK

NORMAL UNIQUE

**\* Here we can see what indexes were created by the Server (UNIQUE). They are always created implicitly when developers specify either PK or UK constraint (but not if DISABLED) and they will be UNIQUE (unless created as DEFERRABLE, and then they will be NONUNIQUE).**

**If we later disable one of these constraints, twin indexes will be dropped then (unless they are NONUNIQUE) and will be recreated again by Server, but that might lock the whole table and cause a huge Database slowdown. \***

SQL> **SHOW USER**

USER is "TOM"

SQL> **desc customer**

Name Null? Type

----------------------------------------- -------- ------------------

CUSTID NOT NULL NUMBER(6)

NAME CHAR(45)

ADDRESS CHAR(40)

CITY CHAR(30)

STATE CHAR(2)

ZIP CHAR(9)

AREA NUMBER(3)

PHONE CHAR(9)

REPID NOT NULL NUMBER(4)

CREDITLIMIT NUMBER(9,2)

**RECIPE FOR “CLEANING TABLE” WITH DUPLICATE KEYS: Case One**

SQL> **INSERT INTO customer (custid,name, city, state,repid)**

**VALUES (901,'BIG SUGAR','NEW YORK','NY',7844);**

1 row created.

SQL> **INSERT INTO customer (custid,name, city, state,repid)**

**VALUES (777,'BIG SUGAR','CHICAGO','IL',7844);**

1 row created.

SQL> **SELECT \* FROM customer**

**WHERE custid > 500;**

CUSTID NAME

---------- ---------------------------------------------

ADDRESS CITY ST

---------------------------------------- -----------------------------

ZIP AREA PHONE REPID CREDITLIMIT

--------- ---------- --------- ---------- -----------

777 **BIG SUGAR**

CHICAGO IL

7844

901 **BIG SUGAR**

NEW YORK NY

7844

SQL> **commit;**

Commit complete.

**\* We just inserted two duplicate values, because there was NO UK constraint placed on column Name in table Customer \***

SQL> **conn / as sysdba**

Connected.

SQL>

SQL> **host**

[oracle@oracledb19c ~]$ **cd $ORACLE\_HOME/rdbms/admin**

[oracle@oracledb19c admin]$ ls -l utle\*

-rw-r--r-- 1 oracle dba 5479 May 29 2017 utledtol.sql

-rw-r--r-- 1 oracle dba 5386 Feb 20 2014 utlenc.sql

-rw-r--r-- 1 oracle dba 15475 May 29 2017 utlestat.sql

-rw-r--r-- 1 oracle dba 1053 May 29 2017 utlexcpt.sql

-rw-r--r-- 1 oracle dba 1099 May 29 2017 utlexpt1.sql

[oracle@oracledb19c admin]$ **cat utlexcpt.sql**

rem

rem $Header: rdbms/admin/utlexcpt.sql /main/4 2017/05/28 22:46:12 stanaya Exp $

rem

Rem Copyright (c) 1991 by Oracle Corporation

Rem NAME

Rem except.sql - <one-line expansion of the name>

Rem DESCRIPTION

Rem <short description of component this file declares/defines>

Rem RETURNS

Rem

Rem NOTES

Rem <other useful comments, qualifications, etc.>

Rem BEGIN SQL\_FILE\_METADATA

Rem SQL\_SOURCE\_FILE: rdbms/admin/utlexcpt.sql

Rem SQL\_SHIPPED\_FILE: rdbms/admin/utlexcpt.sql

Rem SQL\_PHASE: UTILITY

Rem SQL\_STARTUP\_MODE: NORMAL

Rem SQL\_IGNORABLE\_ERRORS: NONE

Rem END SQL\_FILE\_METADATA

Rem

Rem MODIFIED (MM/DD/YY)

Rem traney 04/05/11 - 35209: long identifiers dictionary upgrade

Rem glumpkin 10/20/92 - Renamed from EXCEPT.SQL

Rem epeeler 07/22/91 - add comma

Rem epeeler 04/30/91 - Creation

**create table exceptions(row\_id rowid,**

**owner varchar2(128),**

**table\_name varchar2(128),**

**constraint varchar2(128));**

[oracle@oracledb19c admin]$ **exit**

exit

SQL> **ALTER TABLE tom.customer**

**ADD CONSTRAINT cust\_name\_uk UNIQUE(name)**

**EXCEPTIONS INTO exceptions;**

ALTER TABLE tom.customer

\*

ERROR at line 1:

ORA-01536: space quota exceeded for tablespace 'MINE'

SQL> **ALTER USER tom QUOTA UNLIMITED ON mine;**

User altered.

SQL> **REM Step Zero** -- **Create table EXCEPTIONS in the User Schema**

SQL> **conn tom/cat**

Connected.

SQL> **create table exceptions(row\_id rowid,**

**owner varchar2(128),**

**table\_name varchar2(128),**

**constraint varchar2(128));**

Table created.

SQL> **REM Step One – If constraint is not created yet, try to ADD it and VALIDATE your constraint with EXCEPTIONS table, that will collect duplicate rows.**

SQL> **ALTER TABLE customer**

**ADD CONSTRAINT cust\_name\_uk UNIQUE(name)**

**EXCEPTIONS INTO exceptions;**

ADD CONSTRAINT cust\_name\_uk UNIQUE(name)

\*

ERROR at line 2:

ORA-02299: cannot validate (TOM.CUST\_NAME\_UK) - duplicate keys found

**SQL> REM Step Two -- Show your duplicates that are stored in table Exceptions (Collector table for Duplicate Rows)**

SQL> **SELECT rowid, custid, name**

**FROM customer**

**WHERE rowid IN (SELECT row\_id FROM exceptions**

**WHERE table\_name = 'CUSTOMER');**

ROWID CUSTID NAME

------------------ ---------- ----------------------------------------

AAAU+3AAFAAAAF2AAA 901 BIG SUGAR

AAAU+3AAFAAAAF2AAB 777 BIG SUGAR

**SQL> REM Step Three -- In the meanwhile use NOVALIDATE option that will prevent incoming data from creating duplicates**

SQL> **ALTER TABLE customer ADD CONSTRAINT cust\_name\_uk**

**ENABLE NOVALIDATE;**

ALTER TABLE customer

\*

ERROR at line 1:

ORA-02299: cannot validate (TOM.CUST\_NAME\_UK) - duplicate keys found

**\* Our Constraint was not created before and it is still not there, so we must add it firstly as DISABLED \***

SQL> **ALTER TABLE customer**

**ADD CONSTRAINT cust\_name\_uk UNIQUE(name)**

**DISABLE;**

Table altered.

SQL> **ALTER TABLE customer**

**ENABLE NOVALIDATE CONSTRAINT cust\_name\_uk;**

ALTER TABLE customer

\*

ERROR at line 1:

ORA-02299: cannot validate (TOM.CUST\_NAME\_UK) - duplicate keys found

**\* This means that we miss our twin Index for this UK constraint, and we should have added a Non-Unique Index here for the column Name. That trick will be shown in the next “cleaning example” \***

SQL> **REM Step Four -- Rectify the errors (Update one or more duplicate keys)**

SQL> **UPDATE customer SET name = 'BIGGER SUGAR'**

**WHERE CUSTID = 777;**

1 row updated.

SQL> **COMMIT;**

Commit complete.

SQL> **SELECT \* FROM customer**

**WHERE custid > 500;**

CUSTID NAME

---------- ---------------------------------------------

ADDRESS CITY ST

---------------------------------------- -----------------------------

ZIP AREA PHONE REPID CREDITLIMIT

--------- ---------- --------- ---------- -----------

777 BIGGER SUGAR

CHICAGO IL

7844

901 BIG SUGAR

NEW YORK NY

7844

SQL> **REM Step Five -- Enable your Constraint now**

SQL> **ALTER TABLE customer**

**ENABLE CONSTRAINT cust\_name\_uk;**

Table altered.

SQL> **REM Step Six -- Empty your Collector table**

SQL> **TRUNCATE TABLE exceptions;**

Table truncated.

**LOAD Child/Parent Rows and DEFERRED FK CONSTRAINT**

SQL> **host**

[oracle@oracledb19c ~]$ **pwd**

/home/oracle

[oracle@oracledb19c ~]$ **vi ins\_child\_first.sql**

[oracle@oracledb19c ~]$ **cat ins\***

INSERT INTO emp (empno,ename,deptno) VALUES (501,'JONES',99);

INSERT INTO dept (deptno,dname) VALUES (99,'HR');

COMMIT;

/

**\* This script tries to insert a child row before its parent row. \***

[oracle@oracledb19c ~]$ **exit**

exit

SQL> **show user**

USER is "TOM"

SQL**> @ins\_child\_first;**

INSERT INTO emp (empno,ename,deptno) VALUES (501,'JONES',99)

\*

ERROR at line 1:

ORA-02291: integrity constraint (TOM.EMP\_DEPTNO\_FK) violated - parent key not

found

1 row created.

Commit complete.

Commit complete.

**\* FK constraint in table EMP was created as DEFRRABLE and IMMEDIATE one (sub-default mode) and it will perform line by line check (like for NOT DEFERRABLE one) 🡪 it will ignore "bad" rows, but it will process all other rows that have no errors. Later, you may switch this constraint to DEFERRED mode, and then it will perform only one check at Commit time. In that case, it will process ALL or NONE rows \***

SQL> **SELECT \* FROM dept;**

DEPTNO DNAME LOC

---------- -------------- -------------

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

99 HR

SQL> **DELETE FROM dept**

**WHERE deptno = 99;**

1 row deleted.

SQL> **commit;**

Commit complete.

SQL> **SET CONSTRAINT emp\_deptno\_fk DEFERRED;**

Constraint set.

**\* Manual switch from IMMEDIATE submode to DEFERREED one (possible only for DEFERRABLE constraints) \***

SQL> **@ins\_child\_first;**

1 row created.

1 row created.

Commit complete. **🡪 Check time is delayed till here, and then it will see that a child row has a valid parent row and will process Both rows**

Commit complete.

SQL> **SELECT \* FROM dept;**

DEPTNO DNAME LOC

---------- -------------- -------------

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

99 HR

SQL> **SET CONSTRAINT emp\_deptno\_fk IMMEDIATE;**

Constraint set.

**RECIPE FOR “CLEANING TABLE” WITH DUPLICATE KEYS: Case Two**

SQL> **INSERT INTO dept VALUES (89,'HR',NULL);**

1 row created.

SQL> **COMMIT;**

Commit complete.

SQL> **SELECT \* FROM dept;**

DEPTNO DNAME LOC

---------- -------------- -------------

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

99 HR

89 HR

6 rows selected.

SQL> **ALTER TABLE dept**

**ENABLE CONSTRAINT dept\_dname\_uk**

**EXCEPTIONS INTO exceptions;**

ALTER TABLE dept

\*

ERROR at line 1:

ORA-02299: cannot validate (TOM.DEPT\_DNAME\_UK) - duplicate keys found

SQL> **SELECT \* FROM exceptions;**

ROW\_ID

------------------

OWNER

----------------------------------------------------------------------

TABLE\_NAME

----------------------------------------------------------------------

CONSTRAINT

----------------------------------------------------------------------

AAAU+1AAFAAAAD2AAB

TOM

DEPT

DEPT\_DNAME\_UK

AAAU+1AAFAAAAD2AAC

TOM

DEPT

DEPT\_DNAME\_UK

SQL> **ALTER TABLE dept**

**ENABLE NOVALIDATE CONSTRAINT dept\_dname\_uk;**

ALTER TABLE dept

\*

ERROR at line 1:

ORA-02299: cannot validate (TOM.DEPT\_DNAME\_UK) - duplicate keys found

SQL> **CREATE UNIQUE INDEX dept\_dname\_uk ON dept(dname);**

CREATE UNIQUE INDEX dept\_dname\_uk ON dept(dname)

\*

ERROR at line 1:

ORA-01452: cannot CREATE UNIQUE INDEX; duplicate keys found

SQL> **CREATE INDEX dept\_dname\_uk ON dept(dname);**

Index created.

SQL> **ALTER TABLE dept**

**ENABLE NOVALIDATE CONSTRAINT dept\_dname\_uk;**

Table altered.

SQL> **INSERT INTO dept VALUES (79,'HR',null);**

INSERT INTO dept VALUES (79,'HR',null)

\*

ERROR at line 1:

ORA-00001: unique constraint (TOM.DEPT\_DNAME\_UK) violated

SQL> **SELECT rowid, deptno, dname FROM dept**

**WHERE rowid IN (SELECT row\_id FROM exceptions**

**WHERE table\_name = 'DEPT');**

ROWID DEPTNO DNAME

------------------ ---------- --------------

AAAU+1AAFAAAAD2AAB 99 HR

AAAU+1AAFAAAAD2AAC 89 HR

SQL> **UPDATE dept SET dname = 'OUR HR'**

**WHERE rowid = 'AAAU+1AAFAAAAD2AAC';**

1 row updated.

SQL> **COMMIT;**

Commit complete.

SQL> **SELECT \* FROM dept;**

DEPTNO DNAME LOC

---------- -------------- -------------

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

99 HR

89 OUR HR

6 rows selected.

SQL> **ALTER TABLE dept**

**ENABLE CONSTRAINT dept\_dname\_uk;**

Table altered.

SQL> **TRUNCATE TABLE exceptions;**

Table truncated.

SQL> **ALTER TABLE emp**

**ENABLE CONSTRAINT emp\_sal\_ck;**

ENABLE CONSTRAINT emp\_sal\_ck

\*

ERROR at line 2:

ORA-02293: cannot validate (TOM.EMP\_SAL\_CK) - check constraint violated

SQL> **SELECT \* FROM emp**

**WHERE sal > 4000;**

EMPNO ENAME JOB MGR HIREDATE SAL

---------- ---------- --------- ---------- --------- ---------- ------

COMM DEPTNO

---------------

7839 KING PRESIDENT 17-NOV-81 5000

10

SQL> **UPDATE emp SET sal=4000**

**WHERE empno = 7839;**

1 row updated.

SQL> **commit;**

Commit complete.

SQL> **SELECT \* FROM emp**

**WHERE ename = 'KING';**

EMPNO ENAME JOB MGR HIREDATE SAL

---------- ---------- --------- ---------- --------- ---------- ------

COMM DEPTNO

---------------

7839 KING PRESIDENT 17-NOV-81 4000

10

SQL> **ALTER TABLE emp**

**ENABLE CONSTRAINT emp\_sal\_ck;**

Table altered.

SQL> **EXIT**

Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0