Session 13 Adding /Removing /Viewing TABLE CONSTRAINTS

```
SQL> CONN / as sysdba
Connected.
    * Let's create a new Tablespace INDX just for the index storage *

SQL> CREATE TABLESPACE indx
     DATAFILE '/opt/oracle/oradata/student/indx01.dbf' SIZE 5M;

Tablespace created.

SQL> ALTER USER tom QUOTA 1M ON indx;    → Tom can put his Indexes here
User altered.

SQL> GRANT SELECT_CATALOG_ROLE TO TOM;
Role granted.                         → Tom can browse Data Dictionary

SQL> CONN  tom/cat
Connected.

SQL> SELECT tname FROM TAB;

TNAME
------------------------------------------------------------------
NEW_EMP
BIG_EMP

3 rows selected.

SQL> host cat cr_orders.sql      → script to create and populate 2
                      tables for TOM -- CUSTOMERS and ORDERS

-- Needs Tablespace JOKE
-- SET TERMOUT OFF

DROP TABLE orders
/
DROP TABLE customers
/
CREATE TABLE customers (
cust_code       VARCHAR2(3),
name            VARCHAR2(50),
region          VARCHAR2(5))
TABLESPACE  joke;

CREATE TABLE orders (
ord_id          NUMBER(3),
ord_date        DATE,
cust_code       VARCHAR2(3),
date_of_dely    DATE )
TABLESPACE      joke
```

```
PCTFREE          20
PCTUSED          50
MAXTRANS         100;

INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(610,'11-NOV-
1997','A01');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(611,'15-NOV-
1997','A02');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(612,'19-NOV-
1997','A04');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(601,'05-MAR-
1997','A06');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(602,'09-APR-
1997','A02');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(600,'05-MAR-
1997','A03');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(604,'19-APR-
1997','A06');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(605,'18-MAY-
1997','A06');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(607,'22-MAY-
1997','A04');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(608,'29-MAY-
1997','A04');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(603,'09-APR-
1997','A02');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(613,'06-DEC-
1997','A08');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(614,'06-DEC-
1997','A02');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(616,'08-DEC-
1997','A03');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(619,'27-DEC-
1997','A04');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(617,'10-DEC-
1997','A05');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(615,'06-DEC-
1997','A07');
INSERT INTO  orders(ord_id,ord_date,cust_code) VALUES(618,'20-DEC-
1997','A02');
INSERT INTO  customers VALUES('A01','TKB SPORT SHOP','West');
INSERT INTO  customers VALUES('A02','VOLLYRITE','North');
INSERT INTO  customers VALUES('A03','JUST TENNIS','North');
INSERT INTO  customers VALUES('A04','EVERY MOUNTAIN','South');
INSERT INTO  customers VALUES('A05','SHAPE UP','South');
INSERT INTO  customers VALUES('A06','SHAPE UP','West');
INSERT INTO  customers VALUES('A07','WOMENS SPORTS','South');
INSERT INTO  customers VALUES('A08','NORTH WOODS HEALTH AND FITNESS
SUPPLY CENTER','East');

commit;
```

```
--set termout on

SQL> @cr_orders;           →    run this script as user TOM
DROP TABLE orders
           *
ERROR at line 1:
ORA-00942: table or view does not exist

DROP TABLE customers
           *
ERROR at line 1:
ORA-00942: table or view does not exist

Table created.

Table created.

1 row created.     Etc

Commit complete.

SQL> SELECT * FROM CUSTOMERS;

CUS NAME                                               REGIO
--- -------------------------------------------------- -----
A01 TKB SPORT SHOP                                     West
A02 VOLLYRITE                                          North
A03 JUST TENNIS                                        North
A04 EVERY MOUNTAIN                                     South
A05 SHAPE UP                                           South
A06 SHAPE UP                                           West
A07 WOMENS SPORTS                                      South
A08 NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER       East

8 rows selected.

SQL> SELECT * FROM ORDERS;

    ORD_ID ORD_DATE  CUS DATE_OF_D
---------- --------- --- ---------
       610 11-NOV-97 A01
       611 15-NOV-97 A02
       612 19-NOV-97 A04
       601 05-MAR-97 A06
       602 09-APR-97 A02
       600 05-MAR-97 A03
       604 19-APR-97 A06
       605 18-MAY-97 A06
       607 22-MAY-97 A04
       608 29-MAY-97 A04
       603 09-APR-97 A02
       613 06-DEC-97 A08
       614 06-DEC-97 A02
       616 08-DEC-97 A03
       619 27-DEC-97 A04
       617 10-DEC-97 A05
       615 06-DEC-97 A07
       618 20-DEC-97 A02

18 rows selected.

SQL> host cat  cr_cons.sql  → script to create a 3rd table PRODUCTS and
add constraints to all three tables

-- Login as user who will create these tables (TOM or PAUL etc.)

ALTER TABLE customers
ADD (CONSTRAINT cust_pk PRIMARY KEY(cust_code)
     DEFERRABLE INITIALLY IMMEDIATE
     USING INDEX TABLESPACE indx,
     CONSTRAINT cust_region_ck
        CHECK (region in ('East','West','North','South')))
/

ALTER TABLE  orders
ADD(CONSTRAINT ord_pk PRIMARY KEY(ord_id)
       USING INDEX TABLESPACE indx,
    CONSTRAINT ord_cc_fk FOREIGN KEY(cust_code)
       REFERENCES customers(cust_code)
       DEFERRABLE INITIALLY IMMEDIATE,
    CONSTRAINT ord_dod_ck CHECK (date_of_dely >= ord_date))
/

CREATE TABLE products (
prod_code       NUMBER(6),
description  VARCHAR2(30),
price      NUMBER(8,2),
category   CHAR(2)  )
TABLESPACE joke
/

ALTER TABLE products
ADD CONSTRAINT prod_uk UNIQUE(prod_code)
    DEFERRABLE DISABLE
/

SQL> @cr_cons;             →    run this script as user TOM
Table altered.

Table altered.

Table created.
Table altered.

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 ('CUSTOMERS','ORDERS','PRODUCTS')
     AND      owner = 'TOM'
     ORDER BY 8, 2, 1 ;

CONSTRAINT_NAME                                    C
---------------------------------------------------------
SEARCH_CONDITION
------------------------------------------------------------------------
STATUS   DEFERRABLE     DEFERRED  VALIDATED
-------- -------------- --------- -------------
TABLE_NAME
------------------------------------------------------------------------
CUST_REGION_CK
C
region in ('East','West','North','South')
ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
CUSTOMERS

CUST_PK                                          P
```

```
ENABLED  DEFERRABLE     IMMEDIATE VALIDATED
CUSTOMERS

ORD_DOD_CK                                          C
date_of_dely >= ord_date
ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
ORDERS

ORD_PK                                              P
ENABLED  NOT DEFERRABLE IMMEDIATE VALIDATED
ORDERS

ORD_CC_FK                                           R
ENABLED  DEFERRABLE     IMMEDIATE VALIDATED
ORDERS

PROD_UK                                             U
DISABLED DEFERRABLE     IMMEDIATE NOT VALIDATED
PRODUCTS

6 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> 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, table_name
     FROM       dba_cons_columns
     WHERE      owner = 'TOM'
     AND        table_name IN ('CUSTOMERS', 'ORDERS','PRODUCTS')
     ORDER BY        4, 1;


CONSTRAINT_NAME
------------------------------------------------------------------------
COLUMN_NAME                                       POSITION
------------------------------------------------------------------------
TABLE_NAME
```

```
------------------------------------------------------------------------
CUST_PK
CUST_CODE                                            1
CUSTOMERS

CUST_REGION_CK
REGION
CUSTOMERS

ORD_CC_FK
CUST_CODE                                            1
ORDERS

ORD_DOD_CK
DATE_OF_DELY
ORDERS

ORD_DOD_CK
ORD_DATE
ORDERS

ORD_PK
ORD_ID                                               1
ORDERS

PROD_UK
PROD_CODE                                            1
PRODUCTS


7 rows selected.

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 ('CUSTOMERS','ORDERS','PRODUCTS'));

INDEX_NAME
------------------------------------------------------------------------
INDEX_TYPE                  UNIQUENES
--------------------------- ---------
ORD_PK
NORMAL                      UNIQUE

CUST_PK
NORMAL                      NONUNIQUE
```

* 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.) *

* Here we can see what indexes were created by the Server. 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, 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. *

RECIPE FOR ‟CLEANING TABLE” WITH DUPLICATE KEYS

```
SQL> SELECT * FROM products;
no rows selected

SQL> DESC products
 Name                                      Null?    Type
 ----------------------------------------- -------- --------------------
 PROD_CODE                                  NUMBER(6)
 DESCRIPTION                                        VARCHAR2(30)
 PRICE                                              NUMBER(8,2)
 CATEGORY                                   CHAR(2)

SQL> INSERT INTO products VALUES (2314, 'Soccer Ball', 27.5, 'SP');

1 row created.

SQL> INSERT INTO products VALUES (2314, 'Basket Ball', 29.5, 'SP');

1 row created.

SQL> COMMIT;

Commit complete.
```

* We just inserted two duplicate keys (because our UK constraint was created as Disabled). *

```
SQL> SELECT * FROM products;

 PROD_CODE DESCRIPTION                      PRICE      CA
---------- ------------------------------ ------------------------------
      2314      Soccer Ball                          27.5        SP
      2314      Basket Ball                          29.5        SP

SQL> REM Scenario for CLEANING table with DUPLICATE rows

SQL> REM Step One -- Create table EXCEPTIONS
SQL> @utlexcpt;

Table created.

SQL> DESC exceptions
 Name                         Null?        Type
 ----------------------------------------- ------------
 ROW_ID                                     ROWID
 OWNER                                     VARCHAR2(30)
 TABLE_NAME                                VARCHAR2(30)
 CONSTRAINT                                VARCHAR2(30)

SQL> REM Step Two -- Try to VALIDATE your constraint with EXCEPTIONS
table, that will collect duplicate rows

SQL> ALTER TABLE products
           ENABLE CONSTRAINT prod_uk
           EXCEPTIONS INTO exceptions;
ALTER TABLE products
*
ERROR at line 1:
ORA-02299: cannot validate (TOM.PROD_UK) - duplicate keys found

SQL> SELECT * FROM exceptions;

ROW_ID               OWNER           TABLE_NAME        CONSTRAINT
------------------ ------------------------------ ----------------------
AAABidAAGAAAAFqAAA    TOM                     PRODUCTS        PROD_UK

AAABidAAGAAAAFqAAB    TOM                     PRODUCTS        PROD_UK


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

SQL>  ALTER TABLE products
     ENABLE NOVALIDATE CONSTRAINT prod_uk ;

Table altered.

SQL> INSERT INTO products VALUES (2314,'Golf Ball',3,'RC');
INSERT INTO products VALUES (
*
ERROR at line 1:
ORA-00001: unique constraint (TOM.PROD_UK) violated


SQL> REM Step Four -- Figure out what rows are problematic ones
(collected in the EXCEPTIONS table)

SQL>  SELECT  rowid, prod_code, description
     FROM   products
     WHERE  rowid IN (SELECT row_id FROM exceptions
                      WHERE   table_name = 'PRODUCTS');

ROWID           PROD_CODE  DESCRIPTION
------------------ ---------- ------------------------------------------
-------------
AAABidAAGAAAAFqAAA         2314       Soccer Ball
AAABidAAGAAAAFqAAB         2314       Basket Ball

SQL> REM Step Five -- Rectify the errors (Update one or more duplicate
keuys)

SQL>  UPDATE products
       SET    prod_code = 2315
       WHERE  rowid = 'AAABidAAGAAAAFqAAB';

1 row updated.

SQL> COMMIT;

Commit complete.

SQL> SELECT * FROM products;

 PROD_CODE DESCRIPTION                                PRICE  CA
---------- -------------------------------------------------------------
---------------------
      2314       Soccer Ball                                27.5  SP
      2315       Basket Ball                                29.5  SP

SQL> REM Step Six -- Try to VALIDATE your constraint again (should be
success) and then truncate your collector table

SQL> ALTER TABLE products
     ENABLE CONSTRAINT prod_uk ;

Table altered.

SQL> TRUNCATE TABLE exceptions;

Table truncated.
```

LOAD Child/Parent Rows and DEFERRED FK CONSTRAINT

```
SQL> host  cat ins_ocus1.sql

-- Needs TOM account
SET ECHO ON
INSERT INTO orders VALUES(800,'01-JAN-98','J01',NULL)
/
```

```
INSERT INTO customers VALUES('J01','Sports Unlimited','West')
/

SET ECHO OFF
```

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

```
SQL> @ins_ocus1;

SQL> INSERT INTO orders  VALUES(800,'01-JAN-98','J01',NULL)
/
INSERT INTO system.orders
*
ERROR at line 1:
ORA-02291: integrity constraint (TOM.ORD_CC_FK) violated - parent key
not found

SQL> INSERT INTO customers  VALUES('J01','Sports Unlimited','West')
/
1 row created.

SQL> SET ECHO OFF
```

* FK constraint is created as DEFRRABLE and IMMEDIATE (sub-default mode) and it will perform line by line check (like NOT DEFERRABLE one) → it will ignore "bad" rows, but will process all other rows that have no errors. Later, you may switch this constraint to DEFERRABLE sub-mode and then it will perform only one check at Commit time *

```
SQL> ROLLBACK;

Rollback complete.

SQL> SELECT constraint_name, constraint_type,
            status, deferrable, deferred
           FROM   dba_constraints
           WHERE  table_name = 'ORDERS'
           AND    owner = 'TOM';

CONSTRAINT_NAME      C   STATUS     DEFERRABLE          DEFERRED
------------------------------ - --------- -------------- ---------------
ORD_CC_FK            R   ENABLED    DEFERRABLE          IMMEDIATE
ORD_DOD_CK           C   ENABLED    NOT DEFERRABLE      IMMEDIATE
ORD_PK               P   ENABLED    NOT DEFERRABLE      IMMEDIATE

SQL> SET CONSTRAINT ORD_CC_FK DEFERRED;
Constraint set.
```

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

```
SQL> @ins_ocus1;

SQL> INSERT INTO orders  VALUES(800,'01-JAN-98','J01',NULL)

1 row created.

SQL> INSERT INTO customers VALUES('J01','Sports Unlimited','West')

1 row created.

SQL> SET ECHO OFF

SQL> COMMIT;                → Check time is delayed till here

Commit complete.
```

* FK constraint is now DEFERRED and it will perform check at COMMIT time, that will be successful (because both parent and child row are present in this script). *

```
SQL> SELECT * FROM customers;

CUS   NAME                                                      REGIO
--- -------------------------------------------------------------------
A01   TKB SPORT SHOP                                            West
A02   VOLLYRITE                                                 North
A03   JUST TENNIS                                               North
A04   EVERY MOUNTAIN                                           South
A05   SHAPE UP                                                  South
A06   SHAPE UP                                                  West
A07   WOMENS SPORTS                                             South
A08   NORTH WOODS HEALTH AND FITNESS SUPPLY CENTER               East
J01   Sports Unlimited                                          West
9 rows selected.

SQL> SELECT * FROM orders;

    ORD_ID      ORD_DATE   CUS   DATE_OF_D
----------------------------------------------------------------
       610      11-NOV-97  A01
       611      15-NOV-97  A02
       612      19-NOV-97  A04
       601      05-MAR-97  A06
       602      09-APR-97  A02
       600      05-MAR-97  A03
       604      19-APR-97  A06
       605      18-MAY-97  A06
       607      22-MAY-97  A04
       608      29-MAY-97  A04
       603      09-APR-97  A02
       613      06-DEC-97  A08
       614      06-DEC-97  A02
       616      08-DEC-97  A03
       619      27-DEC-97  A04
       617      10-DEC-97  A05
       615      06-DEC-97  A07
       618      20-DEC-97  A02
       800      01-JAN-98  J01        → new row

19 rows selected.

SQL> SET CONSTRAINT ORD_CC_FK IMMEDIATE;

Constraint set.

SQL> EXIT
```