

Database Programming with SQL 14-3: Managing Constraints Practice Activities

# Objectives

* List four different functions that the ALTER statement can perform on constraints
* Write ALTER TABLE statements to add, drop, disable, and enable constraints
* Name a business function that would require a DBA to drop, enable, and/or disable a constraint or use the CASCADE syntax
* Query the data dictionary for USER\_CONSTRAINTS and interpret the information returned

# Vocabulary

Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **DISABLE CONSTRAINT** | To deactivate an integrity constraint |
| **CASCADE clause** | Disables dependent integrity constraints |
| **ALTER TABLE** | To add, modify, or drop columns from a table |
| **ENABLE CONSTRAINT** | To activate an integrity constraint currently disabled |
| **DROP CONSTRAINT** | Removes a constraint from a table |
| **DROP COLUMN** | Allows user to delete a column from a table |
| **CASCADE CONSTRAINTS** | Defines the actions the database server takes when a user attempts to delete or update a key to which existing foreign keys point |

# Try It / Solve It

Using Oracle Application Express, click the SQL Workshop tab in the menu bar. Click the Object Browser and verify that you have a table named copy\_d\_clients and a table named copy\_d\_events. If you don’t have these tables in your schema, create them before completing the exercises below.

Here is how the original tables are related. The d\_clients table has a primary key client\_number. This has a primary-key constraint and it is referenced in the foreign-key constraint on the d\_events table.

**CREATE TABLE copy\_d\_clients**

**AS ( SELECT \* FROM d\_clients);**

DESCRIBE copy\_d\_clients ;

DESCRIBE d\_clients;

SELECT \* FROM d\_clients ;

SELECT \* FROM copy\_d\_clients ;

SELECT \*

FROM user\_constraints

WHERE LOWER(table\_name) IN ( 'd\_clients', 'copy\_d\_clients');

**CREATE TABLE copy\_d\_events**

**AS ( SELECT \* FROM d\_events);**

DESCRIBE copy\_d\_events ;

DESCRIBE d\_events;

SELECT \* FROM d\_events ;

SELECT \* FROM copy\_d\_events ;

SELECT \*

FROM user\_constraints

WHERE LOWER(table\_name) IN ( 'd\_events', 'copy\_d\_events');

1. What are four functions that an ALTER statement can perform on constraints?

·         **ADD (uses modify clause to add not null on a column though)**

·         **DROP**

·         **ENABLE/DISABLE**

1. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy\_d\_clients table. Name the primary key copy\_d\_clients\_pk . What is the syntax you used to create the PRIMARY KEY constraint to the copy\_d\_clients.table?

**ALTER TABLE copy\_d\_clients**

**ADD  CONSTRAINT copy\_d\_clt\_client\_number\_pk PRIMARY KEY (client\_number);**

SELECT \*

FROM user\_constraints

WHERE LOWER(table\_name) =  'copy\_d\_clients' and constraint\_type = 'P';

1. Create a FOREIGN KEY constraint in the copy\_d\_events table. Name the foreign key copy\_d\_events\_fk. This key references the copy\_d\_clients table client\_number column. What is the syntax you used to create the FOREIGN KEY constraint in the copy\_d\_events table?

**ALTER TABLE  copy\_d\_events**

**ADD CONSTRAINT copy\_d\_eve\_client\_number\_fk FOREIGN KEY (client\_number) REFERENCES  copy\_d\_clients (client\_number) ENABLE;**

SELECT \*

FROM user\_constraints

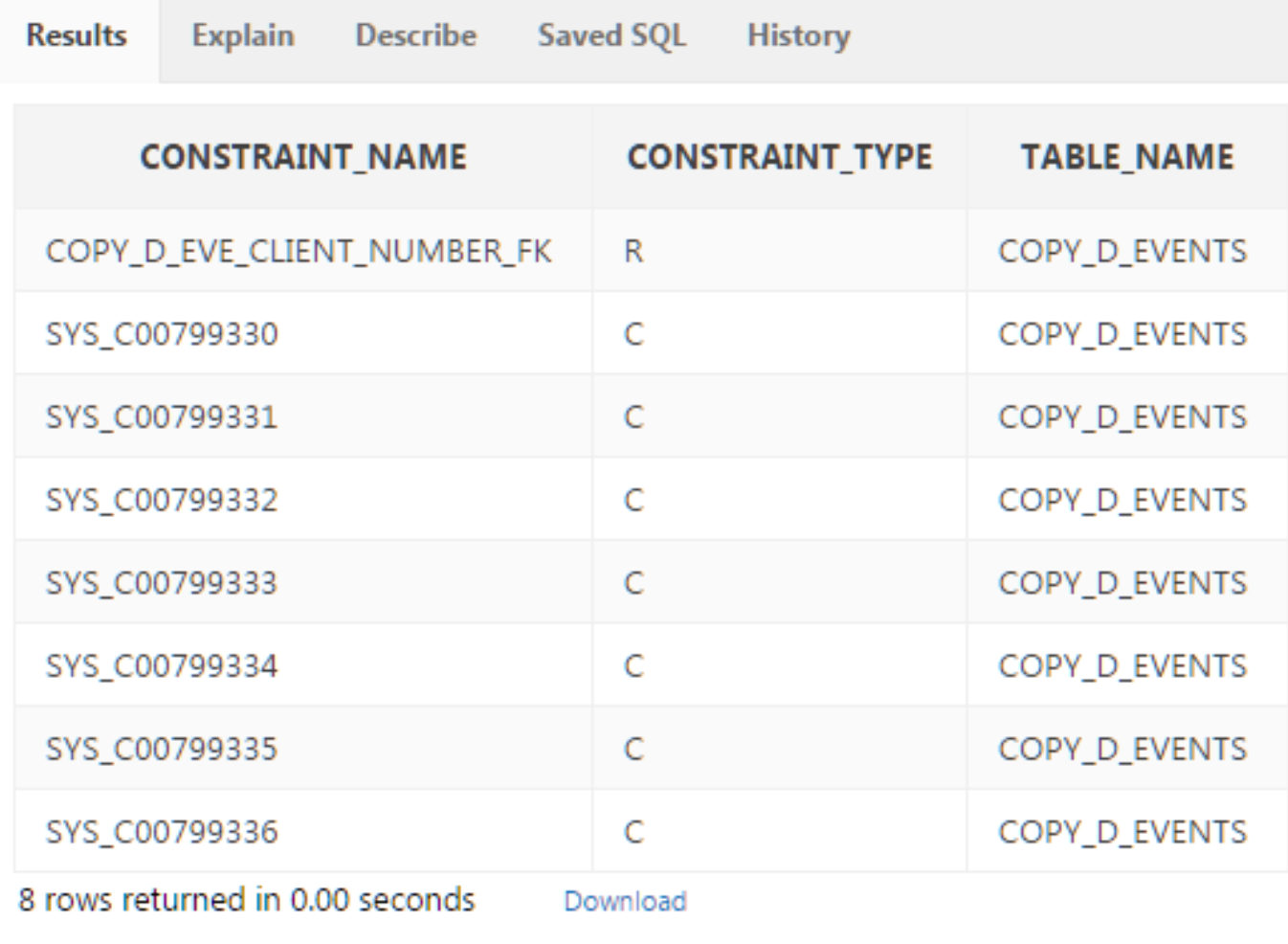
WHERE LOWER(table\_name) =  'copy\_d\_events' and constraint\_type = 'R';

1. Use a SELECT statement to verify the constraint names for each of the tables. Note that the tablenames must be capitalized.

**SELECT constraint\_name, constraint\_type, table\_name**

**FROM user\_constraints**

**WHERE table\_name =  UPPER('copy\_d\_events') ;**



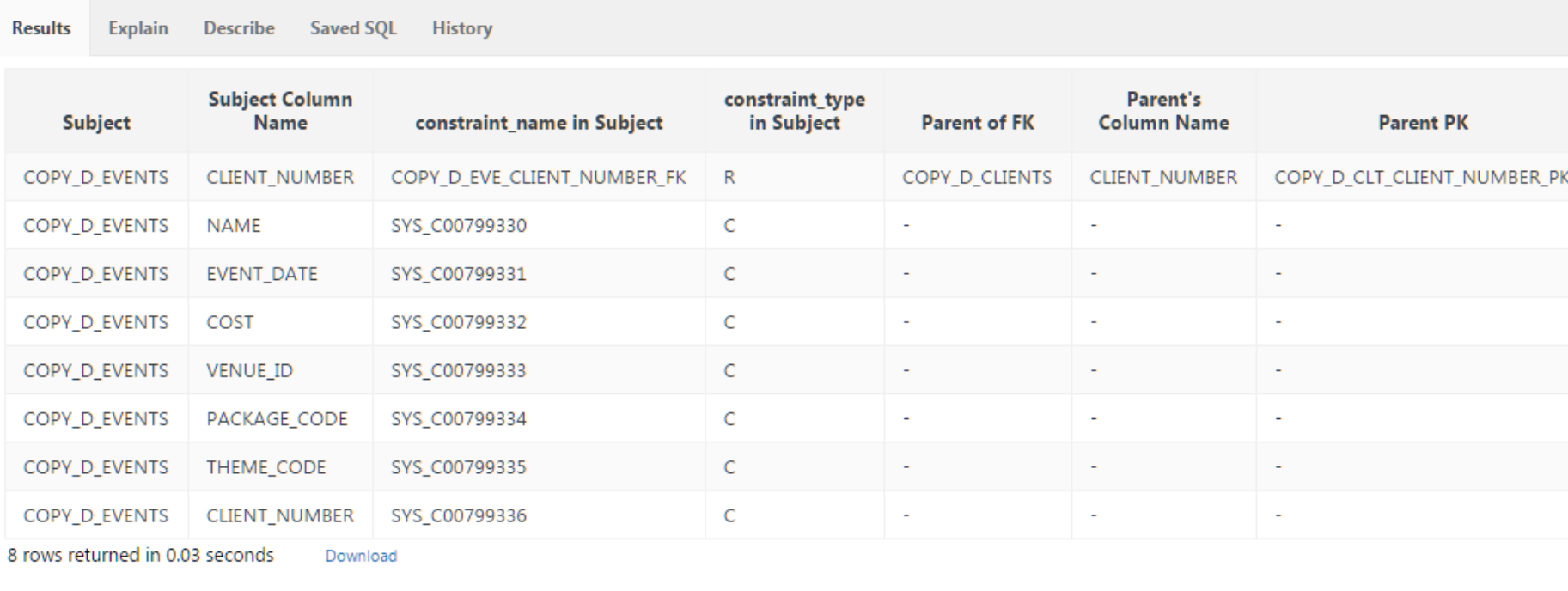
SELECT chld.table\_name "Subject", chldcols.column\_name "Subject Column Name", chld.constraint\_name "constraint\_name in Subject",  chld.constraint\_type  "constraint\_type in Subject",  prnt.table\_name "Parent of FK",  prntcols.column\_name "Parent's Column Name", prnt.constraint\_name "Parent PK"

FROM user\_constraints chld LEFT OUTER JOIN user\_constraints prnt ON  chld.r\_constraint\_name = prnt.constraint\_name

LEFT OUTER JOIN user\_cons\_columns chldcols ON chld.constraint\_name = chldcols.constraint\_name

LEFT OUTER JOIN user\_cons\_columns prntcols ON prnt.constraint\_name = prntcols.constraint\_name

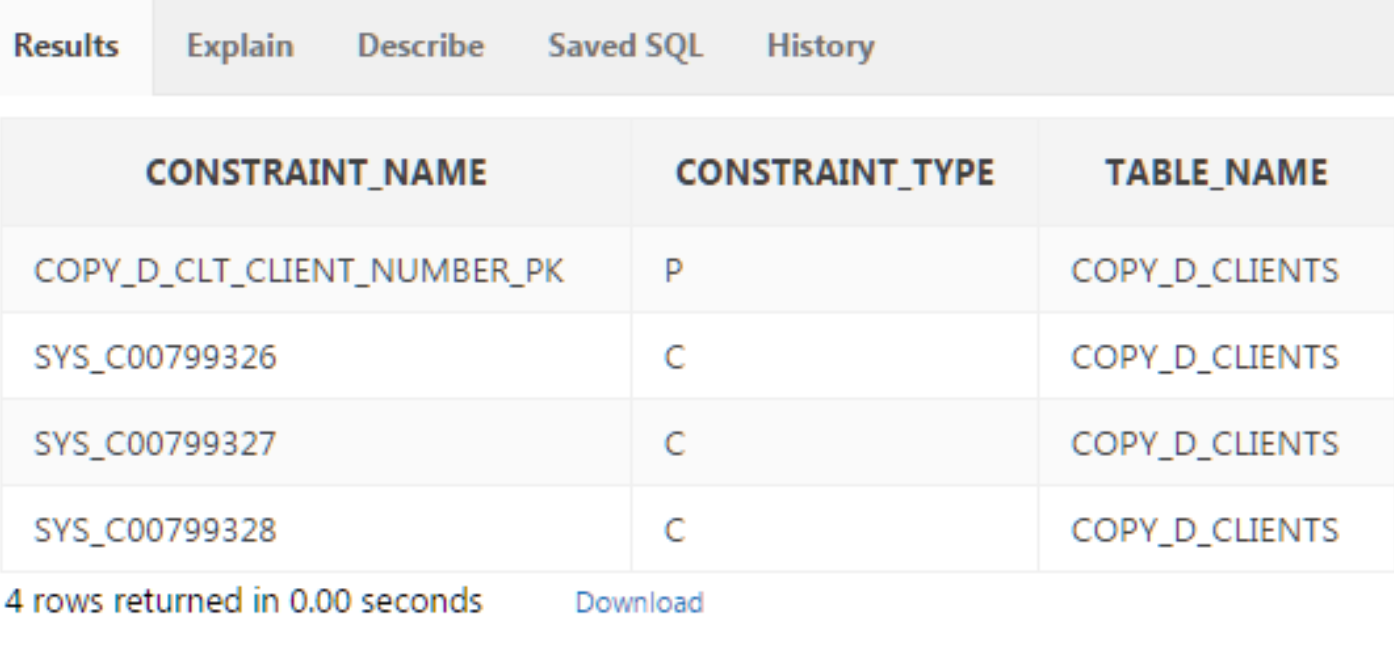
WHERE chld.table\_name = UPPER('copy\_d\_events');



**SELECT constraint\_name, constraint\_type, table\_name**

**FROM user\_constraints**

**WHERE table\_name =  UPPER('copy\_d\_clients') ;**



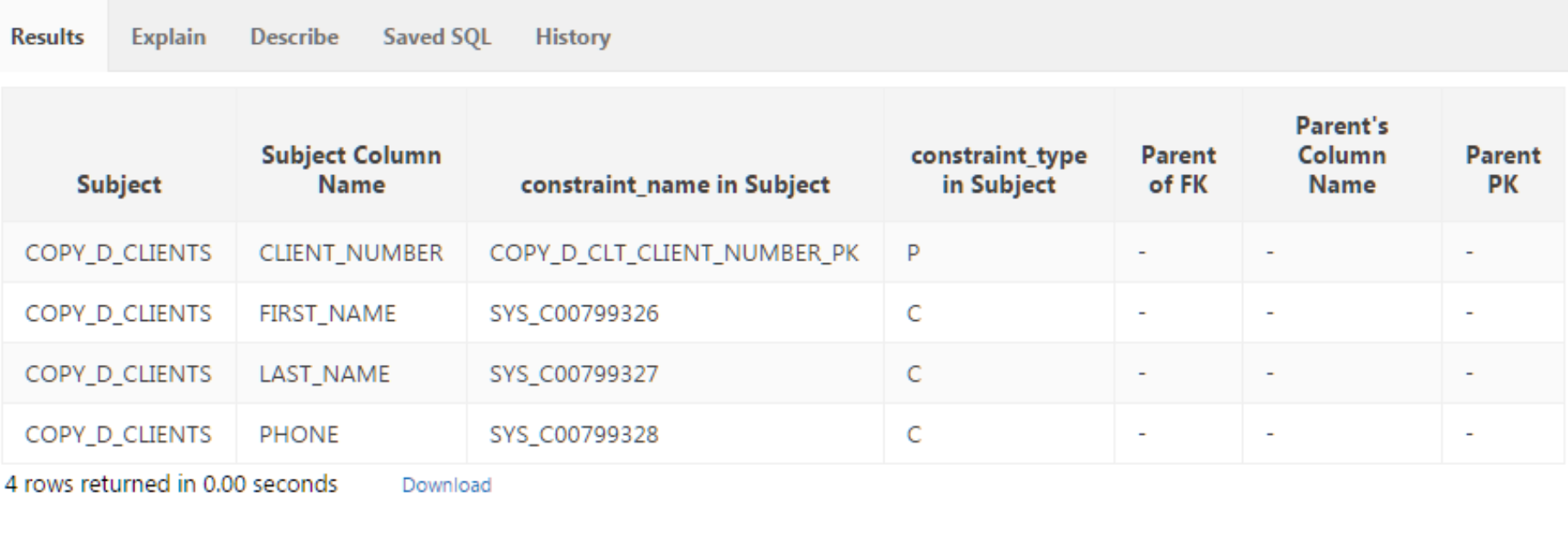
SELECT chld.table\_name "Subject", chldcols.column\_name "Subject Column Name", chld.constraint\_name "constraint\_name in Subject",  chld.constraint\_type  "constraint\_type in Subject",  prnt.table\_name "Parent of FK",  prntcols.column\_name "Parent's Column Name", prnt.constraint\_name "Parent PK"

FROM user\_constraints chld LEFT OUTER JOIN user\_constraints prnt ON  chld.r\_constraint\_name = prnt.constraint\_name

LEFT OUTER JOIN user\_cons\_columns chldcols ON chld.constraint\_name = chldcols.constraint\_name

LEFT OUTER JOIN user\_cons\_columns prntcols ON prnt.constraint\_name = prntcols.constraint\_name

WHERE chld.table\_name = UPPER('copy\_d\_clients');



* 1. The constraint name for the primary key in the copy\_d\_clients table is **COPY\_D\_CLT\_CLIENT\_NUMBER\_PK**.
  2. The constraint name for the foreign key in the copy\_d\_events table is **COPY\_D\_EVE\_CLIENT\_NUMBER\_FK**.

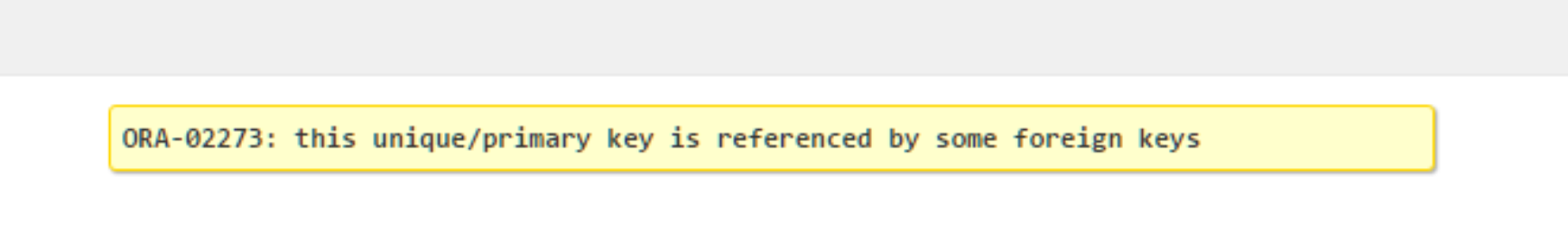
1. Drop the PRIMARY KEY constraint on the copy\_d\_clients table. Explain your results.

ALTER TABLE copy\_d\_clients

DROP CONSTRAINT COPY\_D\_CLT\_CLIENT\_NUMBER\_PK  ;

I get message:

**ORA-02273: this unique/primary key is referenced by some foreign keys**



1. Add the following event to the copy\_d\_events table. Explain your results.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | NAME | EVENT\_DATE | DESCRIPTION | COST | VENUE  \_ID | PACKAGE\_ CODE | THEME\_ CODE | CLIENT\_ NUMBER |
| 140 | Cline Bas Mitzvah | 15-Jul-2004 | Church and Private Home formal | 4500 | 105 | 87 | 77 | 7125 |

INSERT INTO copy\_d\_events(client\_number,id,name,event\_date,description,cost,venue\_id,package\_code,theme\_code)

VALUES(7125,140,'Cline Bas Mitzvah',TO\_DATE('15-Jul-2004','dd-Mon-yyyy'),'Church and Private Home formal',4500,105,87,77);

**ORA-02291: integrity constraint (HKUMAR.COPY\_D\_EVE\_CLIENT\_NUMBER\_FK) violated - parent key not found**

SELECT \* FROM copy\_d\_clients WHERE client\_number = 7125;

no data found

That is why I got above error ORA-02291.

1. Create an ALTER TABLE query to disable the primary key in the copy\_d\_clients table. Then add the values from #6 to the copy\_d\_events table. Explain your results.

**ALTER TABLE copy\_d\_clients**

**DISABLE CONSTRAINT COPY\_D\_CLT\_CLIENT\_NUMBER\_PK ;**

I get message ORA-02297: cannot disable constraint (HKUMAR.COPY\_D\_CLT\_CLIENT\_NUMBER\_PK) - **dependencies exist.**

ALTER TABLE copy\_d\_clients

DISABLE CONSTRAINT COPY\_D\_CLT\_CLIENT\_NUMBER\_PK **CASCADE**;

**Table altered.**

1. Repeat question 6: Insert the new values in the copy\_d\_events table. Explain your results.

**INSERT INTO copy\_d\_events(client\_number,id,name,event\_date,description,cost,venue\_id,package\_code,theme\_code)**

**VALUES(7125,140,'Cline Bas Mitzvah',TO\_DATE('15-Jul-2004','dd-Mon-yyyy'),'Church and Private Home formal',4500,105,87,77);**

**1 row(s) inserted.**

1. Enable the primary-key constraint in the copy\_d\_clients table. Explain your results.

ALTER TABLE copy\_d\_clients

ENABLE CONSTRAINT COPY\_D\_CLT\_CLIENT\_NUMBER\_PK CASCADE;

ORA-00933: SQL command not properly ended

**ALTER TABLE copy\_d\_clients**

**ENABLE CONSTRAINT COPY\_D\_CLT\_CLIENT\_NUMBER\_PK ;**

In case of enable I don’t have option like CASCADE.

1. If you wanted to enable the foreign-key column and reestablish the referential integrity between these two tables, what must be done?

ALTER TABLE copy\_d\_events

ENABLE CONSTRAINT COPY\_D\_EVE\_CLIENT\_NUMBER\_FK;

ORA-02298: cannot validate (HKUMAR.COPY\_D\_EVE\_CLIENT\_NUMBER\_FK) - parent keys not found

So first I need to fix the row with client\_number to a valid value/null.

But since client\_number is not nullable, I will either have to delete invalid row or update that row.

**DELETE FROM copy\_d\_events WHERE**

**client\_number  NOT IN ( SELECT client\_number FROM copy\_d\_clients);**

1 row(s) deleted.

**ALTER TABLE copy\_d\_events**

**ENABLE CONSTRAINT COPY\_D\_EVE\_CLIENT\_NUMBER\_FK;**

Table altered.

1. Why might you want to disable and then re-enable a constraint?

**Generally to make bulk operations fast, where my input data is diligently sanitized and I am sure, it is safe to save some time in this clumsy process.**

1. Query the data dictionary for some of the constraints that you have created. How does the data dictionary identify each constraint type?

**Queries are same as in point 2,3, 4 above.**

**I can check value of CONSTRAINT\_TYPE in all\_constraints/user\_constraints view.**

·         **C - Check constraint**

**Sub-case - if I see SEARCH\_CONDITION something like "FIRST\_NAME" IS NOT NULL , its a NOT NULL constraint.**

·         **P - Primary key**

·         **R - Referential integrity (fk)**

·         **U - Unique key**

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.