

*** EXPERIMENT NO: 02 ***

Aim : To establish a multi-relation database and execute SQL queries involving insertions, deletions and updating on it.

Problem Statement : Establish an environment for executing the queries based on the logical schemata and the database structuring for the **SalesCo** database given below...

CUSTOMER (C_CODE, LNAME, FNAME, C_AREA, C_PHONE, BALANCE)

INVOICE (INV_NUM, c_code, INV_DATE)

LINE (INV_NUM, L_NUM, P_CODE, L_UNITS, L_PRICE)

PRODUCT (P_CODE, DESCRIPT, P_DATE, QTY, P_MIN, P_PRICE, P_DISC, V_CODE)

VENDOR (V_CODE, V_NAME, V_CONTACT, V_AREA, V_PHONE, V_STATE, V_ORDER)

Author : Bhavesh Kewalramani

Roll No : 025 [5A]

Date : 23-July-2021

QUERY-02: For each table of SalesCo database, list all the enforced constraints. (Use USER_CONSTRAINTS view).

```
SELECT TABLE_NAME, CONSTRAINT_NAME, CONSTRAINT_TYPE
2   FROM USER_CONSTRAINTS
3   WHERE TABLE_NAME IN ('CUSTOMER', 'PRODUCT', 'INVOICE', 'LINE', 'VENDOR');
```

TABLE_NAME	CONSTRAINT_NAME	C
VENDOR	VENDOR_CK_V_STATE	C
VENDOR	VENDOR_CK_V_AREA	C
VENDOR	VENDOR_CK_V_CODE	C
VENDOR	SYS_C007405	C
VENDOR	SYS_C007404	C
VENDOR	SYS_C007403	C
VENDOR	SYS_C007402	C
VENDOR	SYS_C007401	C
VENDOR	SYS_C007400	C
VENDOR	SYS_C007399	C

PRODUCT	PRODUCT_CK_P_MIN	C
---------	------------------	---

TABLE_NAME	CONSTRAINT_NAME	C
------------	-----------------	---

-----	-----	-
PRODUCT	SYS_C007416	C
PRODUCT	SYS_C007415	C
PRODUCT	SYS_C007414	C
PRODUCT	SYS_C007413	C
PRODUCT	SYS_C007412	C
PRODUCT	SYS_C007411	C
PRODUCT	SYS_C007410	C
LINE	LINE_CK_L_PRICE	C
LINE	LINE_CK_L_UNITS	C
LINE	LINE_CK_L_NUM	C
LINE	SYS_C007424	C

TABLE_NAME	CONSTRAINT_NAME	C
------------	-----------------	---

-----	-----	-
LINE	SYS_C007423	C
LINE	SYS_C007422	C
LINE	SYS_C007421	C
LINE	SYS_C007420	C
INVOICE	INVOICE_CK_INV_NUM	C
INVOICE	SYS_C007395	C
INVOICE	SYS_C007394	C
INVOICE	SYS_C007393	C
CUSTOMER	CUSTOMER_CK_C_AREA	C
CUSTOMER	CUSTOMER_CK_C_CODE	C
CUSTOMER	SYS_C007389	C

TABLE_NAME	CONSTRAINT_NAME	C
------------	-----------------	---

-----	-----	-
CUSTOMER	SYS_C007388	C
CUSTOMER	SYS_C007387	C
CUSTOMER	SYS_C007386	C
CUSTOMER	SYS_C007385	C
CUSTOMER	SYS_C007384	C
PRODUCT	PRODUCT_VENDOR_FK_V_CODE	R
LINE	LINE_PRODUCT_FK_P_CODE	R
LINE	LINE_INVOICE_FK_INV_NUM	R
INVOICE	INVOICE_CUSTOMER_FK_C_CODE	R
VENDOR	VENDOR_PK_V_CODE	P

PRODUCT	PRODUCT_PK_P_CODE	P
TABLE_NAME	CONSTRAINT_NAME	C
-----	-----	-
LINE	LINE_PK_INV_NUM_L_NUM	P
INVOICE	INVOICE_PK_INV_NUM	P
CUSTOMER	CUSTOMER_PK_C_CODE	P

47 rows selected.

QUERY-03: Write SQL code to insert a LINE record - 1009, 1, HW15X, 20, 15.50.

What are the problems encountered?

Assume that the 40 units of the product "Ball Pin Hammer" were supplied by "Hindustan Forge" located in "KY" at unit price of 17.50 on January 21, 2021. Minimum stock quantity was anticipated to be 10. The line was billed to "You" located in area 904 with phone 3562098 and a balance of 700.00 on July 18, 2021. The supplier with ID 24992 has a contact named "Your Sibling" with phone 2863322.

Write appropriate SELECT statements to showcase the effects of the query.

INSERT INTO LINE

2 VALUES(1009,1,'HW15X',20,15.50);

INSERT INTO LINE

*

ERROR at line 1:

ORA-02291: integrity constraint (CS5025.LINE_PRODUCT_FK_P_CODE) violated -
parent key not found

As P_CODE is a foreign key in LINE and primary key in PRODUCT so 1009 should be stored first in PRODUCT and then it should be used in Line.

INSERT INTO PRODUCT

2 VALUES('HW15X','Ball Pin Hammer','21-Jan-2021',40,10,17.50,0,24992);

```
INSERT INTO PRODUCT
```

```
*
```

```
ERROR at line 1:
```

```
ORA-02291: integrity constraint (CS5025.PRODUCT_VENDOR_FK_V_CODE) violated -  
parent key not found
```

Again while storing record of HW15X in PRODUCT, V_CODE is also not stored in VENDOR and we are using it in PRODUCT. V_CODE being foreign key in PRODUCT and primary key in VENDOR should be stored first in VENDOR and then should be used.

```
INSERT INTO VENDOR
```

```
2 VALUES (24992,'Hindustan Forge','Nancy Kewalramani' ,904,286332,'KY',  
'N');
```

```
1 row created.
```

Now, we can store the records in PRODUCT and LINE

```
INSERT INTO PRODUCT
```

```
2 VALUES ('HW15X','Ball Pin Hammer','21-Jan-2021',40,10,17.50,0,24992);
```

```
1 row created.
```

```
INSERT INTO LINE
```

```
2 VALUES (1009,1,'HW15X',20,17.50);
```

```
INSERT INTO LINE
```

```
*
```

```
ERROR at line 1:
```

```
ORA-02291: integrity constraint (CS5025.LINE_INVOICE_FK_INV_NUM) violated -  
parent key not found
```

Again while storing record in LINE we haven't stored the 1009 in INVOICE as INV_NUM is a foreign key in LINE and primary key in INVOICE and we are using it without storing it.

INSERT INTO INVOICE

2 VALUES(1009,10020,'18-Jul-2021');

INSERT INTO INVOICE

*

ERROR at line 1:

ORA-02291: integrity constraint (CS5025.SYS_C007394) violated -
parent key not found

Also storing record into INVOICE, C_CODE is primary key in CUSTOMER and foreign key in INVOICE so it has to stored first in CUSTOMER and then should be used.

INSERT INTO CUSTOMER

2 VALUES(10020,'Kewalram','Bhavesh',904,3562098,700.00);

1 row created.

Now we try to store record in INVOICE and LINE

INSERT INTO INVOICE

2 VALUES(1009,10020,'18-Jul-2021');

1 row created.

INSERT INTO LINE

2 VALUES(1009,1,'HW15X',20,17.50);

1 row created.

QUERY-04: Write SQL code that will list P_CODE, DESCRIPT, V_CODE for all products that are some kind of hammers or screws or drills.

```
SELECT P_CODE,DESCRIPT,V_CODE
2 FROM PRODUCT
3 WHERE DESCRIPT LIKE '%Hammer' OR DESCRIPT LIKE '%Screw' or DESCRIPT
LIKE '%Drill';
```

P_COD	DESCRIPT	V_CODE
AB112	Power Drill	25595
CD00X	Cordless Drill	25595
CH10X	Claw Hammer	21225
SH100	Sledge Hammer	
MC001	Metal Screw	21225
WC025	2.5in wide Screw	21231
HW15X	Ball Pin Hammer	24992

QUERY-05: Write the SQL code that will list all products which were added to inventory during 2020.

```
SELECT *
2 FROM PRODUCT
3 WHERE EXTRACT(YEAR FROM P_DATE)=2020;
```

P_COD	DESCRIPT	P_DATE	QTY	P_MIN	P_PRICE
CL025	Hrd. Spring 1/4in	15-JAN-20	15	8	39.95
0	23119				

CL050 Hrd. Spring 1/2in	15-JAN-20	23	5	43.99
0 23119				
CD00X Cordless Drill	20-JAN-20	12	5	38.95
.05 25595				
CH10X Claw Hammer	20-JAN-20	23	10	9.95
.1 21225				
SH100 Sledge Hammer	02-JAN-20	8	5	14.4
.05				
HC100 Hicut Chain Saw	07-FEB-20	11	5	256.99
.05 24288				
PP101 PVC Pipe	20-FEB-20	188	75	5.87
0				
P_COD DESCRIPT	P_DATE	QTY	P_MIN	P_PRICE
-----	-----	-----	-----	-----
P_DISC V_CODE				

MC001 Metal Screw	01-MAR-20	172	75	6.99
0 21225				
WC025 2.5in wide Screw	24-FEB-20	237	100	8.45
0 21231				
SM48X Steel Malting Mesh	17-JAN-20	18	5	119.95
.1 25595				

10 rows selected.

QUERY-06: Write SQL code that will list all invoices billed to customers Elena Johnson. Your query must account for combining the FNAME and LNAME attributes while creating and testing the predicate.

```
SELECT I.*
2  FROM INVOICE I JOIN CUSTOMER C
3  ON I.C_CODE=C.C_CODE
4  WHERE C.FNAME LIKE 'Elena' AND C.LNAME LIKE 'Johnson';
```

INV_NUM	C_CODE	INV_DATE
1002	10011	16-JAN-20
1005	10011	17-JAN-20
1008	10011	17-JAN-20

QUERY-08: Write SQL code that will remove the vendor 23119. Explain the problem(s) encountered (if any).

Now, if the policy decision has been to allow such removals from vendor list by removing the depending relation tuples; modify the constraints in PRODUCT table.

On modifying the constraints, remove the said vendor and check the changes in database. *Revert to the database state as before executing this query.*

```
DELETE FROM VENDOR WHERE V_CODE=23119;
```

```
DELETE FROM VENDOR WHERE V_CODE=23119
```

*

ERROR at line 1:

ORA-02292: integrity constraint (CS5025.PRODUCT_VENDOR_FK_V_CODE) violated -
child record found

To delete a record from the VENDOR we first have to disable the constraints. Here first we have to disable the PRODUCT_VENDOR_FK_V_CODE constraint so that there is no dependency while deleting it.

```
ALTER TABLE PRODUCT
```

```
2  DISABLE CONSTRAINT PRODUCT_VENDOR_FK_V_CODE;
```

Table altered.

```
DELETE FROM VENDOR WHERE V_CODE=23119
```

1 row deleted.

Now as Vendor having V_CODE 23119 is deleted so we also have to delete the records of the products that are stored in PRODUCT. Again for that we will disable the constraint and then delete it.

```
DELETE FROM PRODUCT WHERE V_CODE=23119;
```

2 rows deleted.

Now we will enable the constraint that we had disabled before

```
ALTER TABLE PRODUCT
```

```
2  ENABLE CONSTRAINT PRODUCT_VENDOR_FK_V_CODE;
```

Table altered.

Now we will verify that there no record of vendor having V_CODE 23119

```
SELECT *
```

```
2  FROM VENDOR;
```

V_CODE	V_NAME	V_CONTACT	V_AREA

V_PHONE	V_ V		

21225	Bryson, Inc.	Bella Ford	615

2453628 TN Y

21226 SuperLoo, Inc.	Ching Ming	904
3876330 FL N		

21231 GnB Supply	Ted Jones	615
2889922 TN Y		

21344 Gomez Sons	Mark Welder	615
2986363 KY N		

22587 Downing, Inc.	Simon Singh	901
3985333 GA N		

24004 Almeda House	Almeda Brown	615
2787878 TN N		

24288 Justin Stores	Gracy Yu	615
2543333 TN Y		

V_CODE	V_NAME	V_CONTACT	V_AREA

V_PHONE	V_Y		

25443 Super Systems	Ted Hwang	904
3561111 FL N		

25501 Silvermines Ltd.	Anne White	615
2983455 TN N		

25595 HighEnd Supplies	Smith Rust	904
3098773 FL Y		

24992 Hindustan Forge
286332 KY N

Nancy Kewalramani

904

11 rows selected.

```
SELECT *  
2 FROM PRODUCT;
```

P_COD	DESCRIPT	P_DATE	QTY	P_MIN	P_PRICE
AB112	Power Drill	03-NOV-19	8	5	109.99
0	25595				
SB725	7.25in Saw Blade	13-DEC-19	32	15	14.99
.05	21344				
SB900	9.00 in Saw Blade	13-NOV-19	18	12	17.49
0	21344				
JB012	Jigsaw 12in Blade	30-DEC-19	8	5	109.92
.05	24288				
JB008	Jigsaw 8in Blade	24-DEC-19	6	5	99.87
.05	24288				
CD00X	Cordless Drill	20-JAN-20	12	5	38.95
.05	25595				
CH10X	Claw Hammer	20-JAN-20	23	10	9.95
.1	21225				

P_COD	DESCRIPT	P_DATE	QTY	P_MIN	P_PRICE
-------	----------	--------	-----	-------	---------

P_DISC	V_CODE
--------	--------

SH100	Sledge Hammer	02-JAN-20	8	5	14.4
	.05				

RF100	Rat Tail File	15-DEC-19	43	20	4.99
	0 21344				

HC100	Hicut Chain Saw	07-FEB-20	11	5	256.99
	.05 24288				

PP101	PVC Pipe	20-FEB-20	188	75	5.87
	0				

MC001	Metal Screw	01-MAR-20	172	75	6.99
	0 21225				

WC025	2.5in wide Screw	24-FEB-20	237	100	8.45
	0 21231				

SM48X	Steel Malting Mesh	17-JAN-20	18	5	119.95
-------	--------------------	-----------	----	---	--------

P_COD	DESCRIPT	P_DATE	QTY	P_MIN	P_PRICE
-------	----------	--------	-----	-------	---------

P_DISC	V_CODE
--------	--------

.1	25595
----	-------

HW15X	Ball Pin Hammer	21-JAN-21	40	10	17.5
	0 24992				

AB111 Reversible Drill	22-JUL-21	15	5	435
.1 24992				
PP102 PVC Pipe	23-JUL-21	50	12	15.25
.02 24992				

17 rows selected.

QUERY-09: Write SQL code that lists all products that were supplied by vendors belonging to the state 'KY' arranged in increasing sequence of vendor code. The output should include vendor code, vendor's name, product code, product description, vendor contact, and inventory purchase date.

```

SELECT P.V_CODE, V.V_NAME, P.P_CODE, P.DEScript, V.V_PHONE, P.P_DATE
2  FROM VENDOR V JOIN PRODUCT P
3  ON V.V_CODE=P.V_CODE
4  WHERE V.V_STATE='KY'
5  ORDER BY P.V_CODE ASC;

```

V_CODE	V_NAME	P_COD	DESCRIPT
21344	Gomez Sons	SB725	7.25in Saw Blade
2986363	13-DEC-19		
21344	Gomez Sons	SB900	9.00 in Saw Blade
2986363	13-NOV-19		

21344 Gomez Sons	RF100 Rat Tail File
2986363 15-DEC-19	
24992 Hindustan Forge	HW15X Ball Pin Hammer
286332 21-JAN-21	
24992 Hindustan Forge	AB111 Reversible Drill
286332 22-JUL-21	
24992 Hindustan Forge	PP102 PVC Pipe
286332 23-JUL-21	

6 rows selected.

QUERY-10: Write SQL code that will list details of customers who purchased the products CD00X or PP101. The output must include customer name (combination of FName & LName), product code and date of purchase.

```

SELECT DISTINCT C.FNAME, C.LNAME, P.P_CODE, I.INV_DATE
2  FROM CUSTOMER C JOIN INVOICE I
3  ON C.C_CODE=I.C_CODE
4  JOIN LINE L
5  ON I.INV_NUM=L.INV_NUM
6  JOIN PRODUCT P
7  ON L.P_CODE=P.P_CODE
8  WHERE P.P_CODE LIKE 'CD00X' OR P.P_CODE LIKE 'PP101';

```

FNAME	LNAME	P_CODE	INV_DATE
Kathy	Smith	CD00X	16-JAN-20
Elena	Johnson	PP101	17-JAN-20

QUERY-12: Write SQL code that will modify **Query-11**, to include the subtotals for each of the line with invoice numbers. [You are required compute a derived column SUBTOTAL as L_UNITS * L_PRICE].

```

SELECT I.C_CODE, I.INV_NUM, L.L_UNITS, L.L_PRICE, L.L_UNITS*L.L_PRICE AS
SUBTOTAL
2  FROM INVOICE I JOIN LINE L
3  ON I.INV_NUM=L.INV_NUM
4  ORDER BY I.C_CODE ASC, I.INV_NUM DESC;

```

C_CODE	INV_NUM	L_UNITS	L_PRICE	SUBTOTAL
10011	1008	1	9.95	9.95
10011	1008	3	119.95	359.85
10011	1008	5	5.87	29.35
10011	1005	12	5.87	70.44
10011	1002	2	4.99	9.98
10012	1003	1	38.95	38.95
10012	1003	1	39.95	39.95
10012	1003	5	14.99	74.95
10014	1006	3	6.99	20.97
10014	1006	1	256.99	256.99
10014	1006	1	9.95	9.95
10014	1006	1	109.92	109.92
10014	1001	1	14.99	14.99
10014	1001	1	9.95	9.95
10015	1007	2	14.99	29.98
10015	1007	1	4.99	4.99

10018	1004	3	4.99	14.97
10018	1004	2	9.95	19.9
10020	1009	20	17.5	350

19 rows selected.

Conclusion:

In this experiment we established the relation between multiple tables. To connect the tables there should be a common column between the two tables. A column which has unique values and is not null becomes the primary key of the table. This primary key is used as a foreign key in other tables. We can also establish other columns like CHECK, UNIQUE, etc. Now to insert the values in the table containing foreign key, keep in mind to store the record of that foreign key in that table in which it is being used as primary key. To delete the records first disable the constraints and then delete the row and column.

Viva Questions:

1. Bring out differences among super key, candidate key and primary key.

A super key is a single key or a set of several keys that may be used to uniquely identify tuples in a database. Super keys may have redundant characteristics that aren't necessary for identifying tuples.

Candidate keys are a subset of Super keys. They only have the properties that are necessary to uniquely identify tuples. All Candidate keys are Super keys. However, the inverse is not true.

A primary key is a Candidate key that is used to uniquely identify tuples in a database. Primary key values should be distinct and non-null. A table can have numerous Super keys and Candidate keys, but only one Primary key.

2. Differentiate between primary key constraint and unique constraint.

A primary key is a table column that uniquely identifies each tuple (row) in the table. The primary key constrains the table's integrity. In a table, only one primary key can be used. The primary key does not allow duplicate or NULL values. The primary key value in a table changes extremely seldom, therefore it is carefully chosen in cases where changes can occur infrequently. A foreign key from one table can be used to reference a primary key from another table.

Individual tuples in a relation or table are also individually identified by unique key constraints. Unlike the main key, a table can contain many unique

keys. Only one NULL value for a column can be accepted by unique key restrictions. The foreign key of another table refers to unique restrictions as well. It may be used to impose unique constraints on a column or a set of columns that are not main keys.

3. How DROP TABLE differs from TRUNCATE?

The DROP command is used to delete the table definition as well as its contents. The TRUNCATE command, on the other hand, is used to remove all rows from a table.

Table space is removed from memory with the DROP command. The TRUNCATE command, on the other hand, does not release the table space from memory. DROP is a DDL command (Data Definition Language).

TRUNCATE, on the other hand, is a DDL (Data Definition Language) command. The view of table does not exist in the DROP command. A tabular view is available while using this command.

Integrity restrictions will be eliminated using the DROP command. Integrity restrictions will not be removed while in this command.

Undo space is not used in the DROP command. Undo space is used in this command, although it is less than DELETE.

The DROP command is simple to use, but it can lead to problems. While this command is quicker than the DROP command.

4. How does DEFAULT differ from CHECK constraint?

The DEFAULT constraint is used to provide a column's default value. If no alternative value is given, the default value will be applied to all new records. whereas The CHECK constraint is used to restrict the value range that may be entered into a column. A CHECK constraint on a table might limit the values in specific columns based on the values in other columns in the row.