```sql
CREATE TABLE dept (
    deptno INT PRIMARY KEY,
    dept_name VARCHAR(10) CHECK (dept_name IN ('Acc', 'comp', 'elect'))
);
CREATE TABLE emp (
    empno INT PRIMARY KEY,
    emp_name VARCHAR(50) UNIQUE NOT NULL,
    job VARCHAR(10) CHECK (job IN ('Prof', 'AP', 'Lect')),
    sal DECIMAL(10, 2) NOT NULL,
    deptno INT,
    mgr_no INT,
    FOREIGN KEY (deptno) REFERENCES dept(deptno),
    FOREIGN KEY (mgr_no) REFERENCES emp(empno)
);
CREATE TABLE S (
    sno INT PRIMARY KEY,
    sname VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL
);
CREATE TABLE P (
    pno INT PRIMARY KEY,
    pname VARCHAR(50) NOT NULL,
    color VARCHAR(20) NOT NULL
);
```

```sql
CREATE TABLE J (
    jno INT PRIMARY KEY,
    jname VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL
);
CREATE TABLE SPJ (
    sno INT,
    pno INT,
    jno INT,
    qty INT NOT NULL,
    PRIMARY KEY (sno, pno, jno),
    FOREIGN KEY (sno) REFERENCES S(sno),
    FOREIGN KEY (pno) REFERENCES P(pno),
    FOREIGN KEY (jno) REFERENCES J(jno)
);
INSERT INTO dept (deptno, dept_name) VALUES (1, 'Acc');
INSERT INTO dept (deptno, dept_name) VALUES (2, 'comp');
INSERT INTO dept (deptno, dept_name) VALUES (3, 'elect');
select * from dept;
```

```
INSERT INTO emp (empno, emp_name, job, sal, deptno, mgr_no) VALUES (101, 'Alice', 'Prof', 75000, 1, NULL);
INSERT INTO emp (empno, emp_name, job, sal, deptno, mgr_no) VALUES (102, 'Bob', 'AP', 60000, 2, 101);
INSERT INTO emp (empno, emp_name, job, sal, deptno, mgr_no) VALUES (103, 'Charlie', 'Lect', 50000, 2, 102);
INSERT INTO emp (empno, emp_name, job, sal, deptno, mgr_no) VALUES (104, 'David', 'Prof', 80000, 3, NULL);
INSERT INTO emp (empno, emp_name, job, sal, deptno, mgr_no) VALUES (105, 'Eve', 'AP', 62000, 3, 104);
select * from emp;
INSERT INTO S (sno, sname, city) VALUES (201, 'John', 'New York');
INSERT INTO S (sno, sname, city) VALUES (202, 'Emma', 'Los Angeles');
INSERT INTO S (sno, sname, city) VALUES (203, 'Michael', 'Chicago');
INSERT INTO S (sno, sname, city) VALUES (204, 'Sophia', 'Houston');
INSERT INTO S (sno, sname, city) VALUES (205, 'David', 'Phoenix');
select * from s;
INSERT INTO P (pno, pname, color) VALUES (301, 'Bolt', 'Red');
INSERT INTO P (pno, pname, color) VALUES (302, 'Screw', 'Blue');
INSERT INTO P (pno, pname, color) VALUES (303, 'Nut', 'Black');
INSERT INTO P (pno, pname, color) VALUES (304, 'Washer', 'Silver');
INSERT INTO P (pno, pname, color) VALUES (305, 'Gear', 'Gold');
select * from P;
```

```
INSERT INTO J (jno, jname, city) VALUES (401, 'Project A', 'New York');
INSERT INTO J (jno, jname, city) VALUES (402, 'Project B', 'Los Angeles');
INSERT INTO J (jno, jname, city) VALUES (403, 'Project C', 'Chicago');
INSERT INTO J (jno, jname, city) VALUES (404, 'Project D', 'Houston');
INSERT INTO J (jno, jname, city) VALUES (405, 'Project E', 'Phoenix');
select * from J;
INSERT INTO SPJ (sno, pno, jno, qty) VALUES (201, 301, 401, 50);
INSERT INTO SPJ (sno, pno, jno, qty) VALUES (202, 302, 402, 30);
INSERT INTO SPJ (sno, pno, jno, qty) VALUES (203, 303, 403, 40);
INSERT INTO SPJ (sno, pno, jno, qty) VALUES (204, 304, 404, 60);
INSERT INTO SPJ (sno, pno, jno, qty) VALUES (205, 305, 405, 20);
select * from SPJ;
ALTER TABLE emp MODIFY (emp_name NULL);
--ALTER TABLE emp MODIFY (job NULL);
ALTER TABLE emp MODIFY (sal NULL);
--ALTER TABLE emp MODIFY (deptno NULL);
--ALTER TABLE emp MODIFY (mgr_no NULL);
```

```sql
SELECT
    c.constraint_name,
    c.constraint_type,
    cc.column_name
FROM
    user_constraints c
JOIN
    user_cons_columns cc
ON
    c.constraint_name = cc.constraint_name
WHERE
    c.table_name = 'EMP';
```

`c.constraint_name`: This retrieves the name of the constraint. Every constraint (e.g., primary key, foreign key, unique constraint) in the database has a unique name.

`c.constraint_type`: This retrieves the type of the constraint. Oracle uses specific letters to denote constraint types:

- `P` for Primary Key
- `U` for Unique Key
- `R` for Referential Integrity (Foreign Key)
- `C` for Check Constraints
- `N` for Not Null

`cc.column_name`: This retrieves the name of the column(s) that the constraint applies to. For example, if there's a primary key on `empno`, this will show `empno`.

`FROM` Clause:

```sql
FROM
    user_constraints c
```

- `user_constraints`: This is a data dictionary view in Oracle. It contains metadata about all the constraints in the schema (tables, columns, etc.). Each row in this table corresponds to a constraint defined in the schema.
- `c`: This is the alias for `user_constraints`. It's shorthand for referencing the `user_constraints` table in the query.

`JOIN` Clause:

```sql
JOIN
    user_cons_columns cc
ON
    c.constraint_name = cc.constraint_name
```

- `user_cons_columns`: This is another data dictionary view that stores information about which columns are associated with which constraints. It contains details about which columns are involved in primary keys, foreign keys, unique keys, and other constraints.
- `cc`: This is the alias for `user_cons_columns`, just like `c` is for `user_constraints`.
- `ON c.constraint_name = cc.constraint_name`: This is the condition that links the `user_constraints` and `user_cons_columns` views. The `constraint_name` field appears in both tables, so we join them on that field to get a list of columns for each constraint.

`WHERE` Clause:

```sql
WHERE
    c.table_name = 'EMP';
```

- `c.table_name = 'EMP'`: This restricts the results to only those constraints that are related to the `EMP` table. It filters the constraints so that we only see those that apply to the `EMP` table. `c.table_name` refers to the table name where the constraint is defined.

```sql
SELECT constraint_name
FROM user_constraints
WHERE table_name = 'EMP'
  AND constraint_type = 'U';
ALTER TABLE emp DROP CONSTRAINT SYS_C00181416517;

SELECT c.constraint_name
FROM user_constraints c
JOIN user_cons_columns cc
  ON c.constraint_name = cc.constraint_name
WHERE c.table_name = 'EMP'
  AND c.constraint_type = 'R'
  AND cc.column_name = 'DEPTNO';
ALTER TABLE emp DROP CONSTRAINT SYS_C00181416518;
```

```
101 REFERENCES dept(deptno);
102 desc emp;
103
104 SELECT constraint_name
105 FROM user_constraints
106 WHERE table_name = 'DEPT'
107   AND constraint_type = 'C';
108 ALTER TABLE dept DROP CONSTRAINT SYS_C00181422861;
109 desc DEPT;
110
111
112
113
114
115
```

TABLE DEPT

Column	Null?	Type
DEPTNO	NOT NULL	NUMBER
DEPT_NAME	-	VARCHAR2(10)

Download CSV

2 rows selected.

```
117 CREATE TABLE emp_copy AS
118 SELECT * FROM emp WHERE 1=0;
119 select * from emp_copy;
120
121 CREATE TABLE dept_copy AS
122 SELECT deptno AS dept_id, dept_name AS department
123 FROM dept WHERE 1=0;
124
```

```
125 UPDATE emp
126 SET emp_name = 'New_Name',
127     job ='Prof'
128 WHERE empno = 101;
129
130 SELECT deptno FROM dept WHERE dept_name = 'comp';
131 DELETE FROM emp
132 WHERE deptno = (SELECT deptno FROM dept WHERE dept_name = 'comp');
133 select * from emp;
134
135
136
137
```

EMPNO	EMP_NAME	JOB	SAL	DEPTNO	MGR_NO	COMM
101	New_Name	Prof	75000	1	-	0
104	David	Prof	80000	3	-	0
105	Eve	AP	62000	3	104	0

Download CSV

3 rows selected.

```
135 UPDATE emp
136 SET deptno = NULL
137 WHERE deptno = 1;
138 DELETE FROM dept
139 WHERE deptno = 1;
140 SELECT * FROM emp WHERE deptno IS NULL;
141 SELECT * FROM dept WHERE deptno = 1;
142
143
144
145
```

1 row(s) updated.

1 row(s) deleted.

EMPNO	EMP_NAME	JOB	SAL	DEPTNO	MGR_NO	COMM
101	New_Name	Prof	75000	-	-	0

Download CSV

no data found