SELECT get\_employee\_name (1) FROM dual;

CREATE OR REPLACE PROCEDURE print\_greeting\_and\_employee(p\_emp\_id NUMBER) IS

v\_emp\_name VARCHAR2(100);

v\_greeting VARCHAR2(100);

BEGIN

v\_emp\_name := get\_employee\_name(p\_emp\_id);

v\_greeting := get\_greeting;

-- Display the results

DBMS\_OUTPUT.PUT\_LINE('Greeting: ' || v\_greeting);

DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' || v\_emp\_name);

END;

SET SERVEROUTPUT ON;

BEGIN

print\_greeting\_and\_employee(1);

END;

CREATE TABLE employees (

employeeid NUMBER PRIMARY KEY,

employeename VARCHAR2(100)

);

CREATE TABLE devices (

deviceid NUMBER PRIMARY KEY,

employeeid NUMBER,

devicename VARCHAR2(100),

CONSTRAINT fk\_employee

FOREIGN KEY (employeeid)

REFERENCES employees(employeeid)

);

INSERT INTO employees (employeeid, employeename) VALUES (1, 'Ali Alqahtani');

INSERT INTO employees (employeeid, employeename) VALUES (2, 'Sara Almutairi');

INSERT INTO employees (employeeid, employeename) VALUES (3, 'Fahad Alharbi');

INSERT INTO employees (employeeid, employeename) VALUES (4, 'Noura Alshammari');

INSERT INTO employees (employeeid, employeename) VALUES (5, 'Hassan Alotaibi');

INSERT INTO employees (employeeid, employeename) VALUES (6, 'Laila Alanzi');

INSERT INTO employees (employeeid, employeename) VALUES (7, 'Omar Alshehri');

INSERT INTO employees (employeeid, employeename) VALUES (8, 'Rania Aljohani');

INSERT INTO employees (employeeid, employeename) VALUES (9, 'Majed Alzahrani');

INSERT INTO employees (employeeid, employeename) VALUES (10, 'Dina Almarri');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (101, 1, 'Laptop - Dell XPS');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (102, 2, 'Smartphone - iPhone 14');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (103, 3, 'Tablet - iPad Pro');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (104, 4, 'Laptop - HP Spectre');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (105, 5, 'Desktop - Lenovo ThinkCentre');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (106, 6, 'Smartwatch - Samsung Galaxy Watch');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (107, 7, 'Phone - Galaxy S23');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (108, 8, 'Laptop - MacBook Air');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (109, 9, 'Printer - Canon PIXMA');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (110, 10, 'Scanner - Epson V600');

DROP TABLE devices;

CREATE TABLE devices (

deviceid NUMBER PRIMARY KEY,

employeeid NUMBER,

devicename VARCHAR2(100),

CONSTRAINT fk\_employee

FOREIGN KEY (employeeid)

REFERENCES employees(employeeid)

);

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (201, NULL, 'Router - TP-Link AX1800');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (202, NULL, 'Monitor - LG UltraFine');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (203, NULL, 'Keyboard - Logitech MX Keys');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (204, NULL, 'Mouse - Razer DeathAdder');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (205, NULL, 'External Drive - WD 1TB');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (301, 1, 'Headset - Bose QC35');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (302, 3, 'Docking Station - Dell WD19');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (303, 5, 'Webcam - Logitech C920');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (304, 7, 'Microphone - Blue Yeti');

INSERT INTO devices (deviceid, employeeid, devicename) VALUES (305, 9, 'Graphics Tablet - Wacom Intuos');

select \* from employees

commit;

select \* from devices

/

SELECT \*

FROM

devices d

INNER JOIN

employees e ON e.employeeid = d.employeeid;

/

SELECT \*

FROM

employees e

LEFT JOIN

devices d ON e.employeeid = d.employeeid;

/

SELECT \*

FROM

employees e

RIGHT JOIN

devices d ON e.employeeid = d.employeeid;

/

SELECT \*

FROM

employees e

LEFT JOIN

devices d ON e.employeeid = d.employeeid

where d.deviceid is null;

/

SELECT \*

FROM

employees e

RIGHT JOIN

devices d ON e.employeeid = d.employeeid;

/

SELECT \*

FROM

employees e

FULL JOIN

devices d ON e.employeeid = d.employeeid;

/

create table store(

storeid NUMBER PRIMARY KEY,

deviceid NUMBER,

storename VARCHAR2(100),

CONSTRAINT fk\_deviceid

FOREIGN KEY (deviceid)

REFERENCES devices(deviceid)

);

create table country(

countryid NUMBER PRIMARY KEY,

countryname VARCHAR2(100),

storeid NUMBER,

CONSTRAINT fk\_storeid

FOREIGN KEY (storeid)

REFERENCES store(storeid)

);

create table manag(

managmentid NUMBER PRIMARY KEY,

managername VARCHAR2(100),

employeeid NUMBER,

CONSTRAINT fk\_employee\_manag FOREIGN KEY (employeeid) REFERENCES employees(employeeid)

);

INSERT INTO store (storeid, deviceid, storename) VALUES (1, 201, 'Store A');

INSERT INTO store (storeid, deviceid, storename) VALUES (2, 202, 'Store B');

INSERT INTO store (storeid, deviceid, storename) VALUES (3, 301, 'Store C');

INSERT INTO store (storeid, deviceid, storename) VALUES (4, 303, 'Store D');

INSERT INTO store (storeid, deviceid, storename) VALUES (5, 305, 'Store E');

commit;

INSERT INTO country (countryid, countryname, storeid) VALUES (1, 'Saudi Arabia', 1);

INSERT INTO country (countryid, countryname, storeid) VALUES (2, 'UAE', 2);

INSERT INTO country (countryid, countryname, storeid) VALUES (3, 'Oman', 3);

INSERT INTO country (countryid, countryname, storeid) VALUES (4, 'Kuwait', 4);

INSERT INTO country (countryid, countryname, storeid) VALUES (5, 'Qatar', 5);

commit;

INSERT INTO manag (managmentid, managername, employeeid) VALUES (1, 'Huda Alharthi', 1);

INSERT INTO manag (managmentid, managername, employeeid) VALUES (2, 'Faisal Alshahrani', 3);

INSERT INTO manag (managmentid, managername, employeeid) VALUES (3, 'Mona Alotaibi', 5);

INSERT INTO manag (managmentid, managername, employeeid) VALUES (4, 'Tariq Alsuwailem', 7);

INSERT INTO manag (managmentid, managername, employeeid) VALUES (5, 'Salem Alqahtani', 9);

commit;

select \*

from employees e

JOIN devices d ON e.employeeid = d.employeeid

JOIN store s ON d.deviceid = s.deviceid

JOIN country c ON s.storeid = c.storeid

JOIN manag m ON m.employeeid = e.employeeid

where e.employeeid = :para;

select \*

from employees e

JOIN devices d ON e.employeeid = d.employeeid

JOIN store s ON d.deviceid = s.deviceid

JOIN country c ON s.storeid = c.storeid

JOIN manag m ON m.employeeid = e.employeeid

where (:para IS NULL OR e.employeeid = :para);

SELECT \*

from employees e

join devices d ON e.employeeid = d.employeeid

join store s ON d.deviceid = s.deviceid

join country c ON s.storeid = c.storeid

join manag m ON m.employeeid = e.employeeid

where

(:employe\_id IS NULL OR e.employeeid = :employe\_id)

AND (:devic\_id IS NULL OR d.deviceid = :devic\_id)

AND (:store\_id IS NULL OR s.storeid = :store\_id);

/

DECLARE

msgpring VARCHAR2(50);

BEGIN

msgpring := 'hello';

DBMS\_OUTPUT.PUT\_LINE(msgpring);

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('error');

END;

/

DROP function func1;

CREATE OR REPLACE PROCEDURE func1

IS

functionmsg VARCHAR2(20);

BEGIN

functionmsg := 'hello';

DBMS\_OUTPUT.PUT\_LINE(functionmsg);

END;

/

EXEC func1;

/

CREATE OR REPLACE PROCEDURE hifunction

IS

himsg VARCHAR2(20);

BEGIN

himsg := 'hi';

DBMS\_OUTPUT.PUT\_LINE(himsg);

END;

/

EXEC hifunction;

/

SELECT OBJECT\_NAME, OBJECT\_TYPE

FROM USER\_PROCEDURES

WHERE OBJECT\_TYPE IN ('PROCEDURE', 'FUNCTION');

/

CREATE FUNCTION get\_greeting

RETURN VARCHAR2 IS

BEGIN

RETURN 'Hello from a function!';

END;

/

BEGIN

DBMS\_OUTPUT.PUT\_LINE(get\_greeting);

END;

/

DROP function get\_greeting;

CREATE OR REPLACE FUNCTION get\_employee\_name(p\_emp\_id NUMBER)

RETURN VARCHAR2 IS

v\_name VARCHAR2(100);

BEGIN

SELECT employeename

INTO v\_name

FROM employees

WHERE employeeid = p\_emp\_id;

RETURN v\_name;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'No employee found with ID ' || p\_emp\_id;

WHEN OTHERS THEN

RETURN 'Error: ' || SQLERRM;

END;

/

BEGIN

DBMS\_OUTPUT.PUT\_LINE(get\_employee\_name(1));

END;

/

CREATE OR REPLACE PROCEDURE alltasks

IS

v\_name VARCHAR2(100);

BEGIN

hifunction;

func1;

v\_name := get\_employee\_name(2);

DBMS\_OUTPUT.PUT\_LINE(v\_name);

END;

/

BEGIN

alltasks;

END;

select