//1

CREATE TABLE employee (

employee\_id NUMBER PRIMARY KEY,

first\_name VARCHAR2(50),

last\_name VARCHAR2(50),

last\_modified TIMESTAMP

);

CREATE OR REPLACE TRIGGER trigger\_updated\_last\_modified

BEFORE UPDATE ON employee

FOR EACH ROW

BEGIN

:NEW.last\_modified := SYSTIMESTAMP;

END;

SELECT employee\_id, first\_name, EXTRACT(HOUR FROM last\_modified) AS hour\_of\_last\_modified

FROM employee;

//2

CREATE TABLE orders (

order\_id NUMBER PRIMARY KEY,

order\_date TIMESTAMP,

order\_amount NUMBER

);

select \* from orders

update orders

set order\_id = 2

where order\_amount = 1;

CREATE OR REPLACE TRIGGER trigger\_prevent\_update\_orders

BEFORE UPDATE ON orders

FOR EACH ROW

BEGIN

IF EXTRACT(HOUR FROM :NEW.order\_date) < 9 OR EXTRACT(HOUR FROM :NEW.order\_date) > 21 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Order can not be updated right now! Please comeback later.');

END IF;

END;

//3

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

first\_name VARCHAR2(50),

last\_name VARCHAR2(50)

);

CREATE TABLE history\_log (

employee\_id NUMBER,

first\_name VARCHAR2(50),

last\_name VARCHAR2(50),

deleted\_at TIMESTAMP DEFAULT SYSTIMESTAMP

);

CREATE OR REPLACE TRIGGER trigger\_history\_log\_employees

BEFORE DELETE ON employees

FOR EACH ROW

BEGIN

INSERT INTO history\_log (employee\_id, first\_name, last\_name, deleted\_at)

VALUES (:OLD.employee\_id, :OLD.first\_name, :OLD.last\_name, SYSTIMESTAMP);

END;

drop table departments

//4

CREATE TABLE departments (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50)

);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

first\_name VARCHAR2(50),

last\_name VARCHAR2(50),

department\_id NUMBER,

CONSTRAINT fk\_dept FOREIGN KEY (department\_id)

REFERENCES departments (department\_id)

);

CREATE OR REPLACE TRIGGER trigger\_prevent\_dept\_delete

BEFORE DELETE ON departments

FOR EACH ROW

DECLARE

child\_count NUMBER;

BEGIN

SELECT COUNT(\*)

INTO child\_count

FROM employees

WHERE department\_id = :OLD.department\_id;

IF child\_count > 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Cannot delete department with existing employees.');

END IF;

END;

//5

CREATE TABLE products (

product\_id NUMBER PRIMARY KEY,

product\_name VARCHAR2(50)

);

CREATE OR REPLACE TRIGGER trigger\_check\_duplicate

BEFORE INSERT OR UPDATE ON products

FOR EACH ROW

DECLARE

duplicate\_count NUMBER;

BEGIN

SELECT COUNT(\*)

INTO duplicate\_count

FROM products

WHERE product\_name = :NEW.product\_name

AND product\_id != :NEW.product\_id;

IF duplicate\_count > 0 THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Duplicate product\_name is not allowed.');

END IF;

END;