

Excerise: 13

231801175

The screenshot shows the OneCompiler web IDE interface. The editor contains a PL/SQL script for a database schema and a trigger. The script defines a 'Parent' table with 'parent_id' as the primary key and 'parent_name' as a VARCHAR2(100). It also defines a 'Child' table with 'child_id' as the primary key, 'parent_id' as a foreign key, and 'child_name' as a VARCHAR2(100). A trigger named 'prevent_parent_deletion' is created, which fires before a delete operation on the 'Parent' table. The trigger declares a variable 'v_child_count' and a cursor to count the number of child records for the parent being deleted. If the count is greater than 0, it raises an application error with the message 'Cannot delete parent record; child records exist.'.

```
1 CREATE TABLE Parent (  
2   parent_id NUMBER PRIMARY KEY,  
3   parent_name VARCHAR2(100)  
4 );  
5  
6 CREATE TABLE Child (  
7   child_id NUMBER PRIMARY KEY,  
8   parent_id NUMBER,  
9   child_name VARCHAR2(100),  
10  FOREIGN KEY (parent_id) REFERENCES Parent(parent_id)  
11 );  
12 CREATE OR REPLACE TRIGGER prevent_parent_deletion  
13 BEFORE DELETE ON Parent  
14 FOR EACH ROW  
15 DECLARE  
16   v_child_count NUMBER;  
17 BEGIN  
18   -- Count the number of child records referencing the parent  
19   SELECT COUNT(*)  
20   INTO v_child_count  
21   FROM Child  
22   WHERE parent_id = :OLD.parent_id;  
23  
24   -- If there are child records, raise an exception  
25   IF v_child_count > 0 THEN  
26     RAISE_APPLICATION_ERROR(-20001, 'Cannot delete parent record; child records exist.');
```

231801175

The screenshot shows the OneCompiler web IDE interface. The editor contains a PL/SQL script for a database schema and a trigger. The script defines an 'Employees' table with 'employee_id' as the primary key, 'email' as a VARCHAR2(100), and 'employee_name' as a VARCHAR2(100). A trigger named 'prevent_duplicate_email' is created, which fires before an insert or update operation on the 'Employees' table. The trigger declares a variable 'v_email_count' and a cursor to count the number of employees with the same email as the one being inserted or updated. If the count is greater than 0, it raises an application error with the message 'Duplicate email found: ' || :NEW.email;.

```
1  
2 CREATE TABLE Employees (  
3   employee_id NUMBER PRIMARY KEY,  
4   email VARCHAR2(100),  
5   employee_name VARCHAR2(100)  
6 );  
7 CREATE OR REPLACE TRIGGER prevent_duplicate_email  
8 BEFORE INSERT OR UPDATE ON Employees  
9 FOR EACH ROW  
10 DECLARE  
11   v_email_count NUMBER;  
12 BEGIN  
13   -- Count how many times the new email appears in the table  
14   SELECT COUNT(*)  
15   INTO v_email_count  
16   FROM Employees  
17   WHERE email = :NEW.email;  
18  
19   -- If the email already exists, raise an exception  
20   IF v_email_count > 0 THEN  
21     RAISE_APPLICATION_ERROR(-20002, 'Duplicate email found: ' || :NEW.email);  
22   END IF;  
23 END;  
24 /
```

231801175

Excerise: 13

The screenshot shows the OneCompiler website interface. The browser tabs include WhatsApp, ChatGPT | OpenAI, ChatGPT, and 42u9vqvgw - Oracle PL/SQL - C. The URL bar shows onecompiler.com/plsql/42u9vqvgw. The main content area displays a SQL script in a text editor. The script creates two tables: 'employees' and 'emp'. The 'employees' table has columns: employee_id (NUMBER, PRIMARY KEY), name (VARCHAR2(100)), salary (NUMBER), and department_id (NUMBER). The 'emp' table has columns: audit_id (NUMBER, GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY), employee_id (NUMBER), old_salary (NUMBER), new_salary (NUMBER), old_department_id (NUMBER), new_department_id (NUMBER), and change_date (TIMESTAMP DEFAULT SYSTIMESTAMP). A trigger named 'trg_audit_employees' is created, which fires after an update on the 'employees' table. The trigger logic logs salary changes and department changes into the 'emp' table. The right sidebar shows the 'STDIN' section with 'Input for the program (Optional)' and the 'Output' section, which displays 'Program did not output anything!'. The Windows taskbar at the bottom shows the search bar and various application icons.

```
1 CREATE TABLE employees (  
2   employee_id NUMBER PRIMARY KEY,  
3   name VARCHAR2(100),  
4   salary NUMBER,  
5   department_id NUMBER  
6 );  
7  
8 CREATE TABLE emp (  
9   audit_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
10  employee_id NUMBER,  
11  old_salary NUMBER,  
12  new_salary NUMBER,  
13  old_department_id NUMBER,  
14  new_department_id NUMBER,  
15  change_date TIMESTAMP DEFAULT SYSTIMESTAMP  
16 );  
17 CREATE OR REPLACE TRIGGER trg_audit_employees  
18 AFTER UPDATE OF salary, department_id ON employees  
19 FOR EACH ROW  
20 BEGIN  
21   -- Log salary change  
22   IF :OLD.salary != :NEW.salary THEN  
23     INSERT INTO emp (employee_id, old_salary, new_salary, old_department_id, new_department_id,  
24                     VALUES (:OLD.employee_id, :OLD.salary, :NEW.salary, :OLD.department_id, :NEW.department_id,  
25                             :NEW.change_date))  
26   END IF;  
27   -- Log department_id change  
28   IF :OLD.department_id != :NEW.department_id THEN  
29     INSERT INTO emp (employee_id, old_salary, new_salary, old_department_id, new_department_id,  
30                     VALUES (:OLD.employee_id, :OLD.salary, :NEW.salary, :OLD.department_id, :NEW.department_id,  
31                             :NEW.change_date))  
32   END IF;  
33
```

The screenshot shows the OneCompiler website interface. The browser tabs include WhatsApp, ChatGPT | OpenAI, ChatGPT, and 42u9vqvgw - Oracle PL/SQL - C. The URL bar shows onecompiler.com/plsql/42u9vqvgw. The main content area displays a SQL script in a text editor. The script creates two triggers: 'trg_audit_employees_update' and 'trg_audit_employees_delete'. The 'trg_audit_employees_update' trigger fires after an update on the 'employees' table and logs the update into the 'audit_log' table. The 'trg_audit_employees_delete' trigger fires after a delete on the 'employees' table and logs the delete into the 'audit_log' table. The right sidebar shows the 'STDIN' section with 'Input for the program (Optional)' and the 'Output' section, which displays 'Program did not output anything!'. The Windows taskbar at the bottom shows the search bar and various application icons.

```
28 /  
29 CREATE OR REPLACE TRIGGER trg_audit_employees_update  
30 AFTER UPDATE ON employees  
31 FOR EACH ROW  
32 BEGIN  
33   INSERT INTO audit_log (table_name, operation, old_values, new_values, user_name)  
34   VALUES ('employees', 'UPDATE',  
35           'employee_id: ' || :OLD.employee_id ||  
36           ', name: ' || :OLD.name ||  
37           ', salary: ' || :OLD.salary ||  
38           ', department_id: ' || :OLD.department_id ||  
39           ', employee_id: ' || :NEW.employee_id ||  
40           ', name: ' || :NEW.name ||  
41           ', salary: ' || :NEW.salary ||  
42           ', department_id: ' || :NEW.department_id ||  
43           ', USER);  
44 END;  
45 /  
46 CREATE OR REPLACE TRIGGER trg_audit_employees_delete  
47 AFTER DELETE ON employees  
48 FOR EACH ROW  
49 BEGIN  
50   INSERT INTO audit_log (table_name, operation, old_values, user_name)  
51   VALUES ('employees', 'DELETE',  
52           'employee_id: ' || :OLD.employee_id ||  
53           ', name: ' || :OLD.name ||  
54           ', salary: ' || :OLD.salary ||  
55           ', department_id: ' || :OLD.department_id ||  
56           ', USER);  
57 END;  
58 /  
59
```

Excerise: 13

231801175

The screenshot shows the OneCompiler web interface with a browser window open to `onecompiler.com/plsql/42u9xhvf`. The editor contains the following SQL code:

```
27 END;
28 /
29 CREATE OR REPLACE TRIGGER trg_audit_employees_update
30 AFTER UPDATE ON employees
31 FOR EACH ROW
32 BEGIN
33     INSERT INTO audit_log (table_name, operation, old_values, new_values, user_name)
34     VALUES ('employees', 'UPDATE',
35            'employee_id: ' || :OLD.employee_id ||
36            ', name: ' || :OLD.name ||
37            ', salary: ' || :OLD.salary ||
38            ', department_id: ' || :OLD.department_id,
39            'employee_id: ' || :NEW.employee_id ||
40            ', name: ' || :NEW.name ||
41            ', salary: ' || :NEW.salary ||
42            ', department_id: ' || :NEW.department_id,
43            USER);
44 END;
45 /
46 CREATE OR REPLACE TRIGGER trg_audit_employees_delete
47 AFTER DELETE ON employees
48 FOR EACH ROW
49 BEGIN
50     INSERT INTO audit_log (table_name, operation, old_values, user_name)
51     VALUES ('employees', 'DELETE',
52            'employee_id: ' || :OLD.employee_id ||
53            ', name: ' || :OLD.name ||
54            ', salary: ' || :OLD.salary ||
55            ', department_id: ' || :OLD.department_id,
56            USER);
57 END;
58 /
```

The right-hand pane shows the execution output, which is empty, displaying "Program did not output anything!". The Windows taskbar at the bottom shows the time as 11:31 on 06-10-2024.

The screenshot shows the OneCompiler web interface with a browser window open to `onecompiler.com/plsql/42u9xhvf`. The editor contains the following SQL code:

```
1 CREATE TABLE sales (
2     sale_id NUMBER PRIMARY KEY,
3     amount NUMBER,
4     running_total NUMBER
5 );
6 CREATE OR REPLACE TRIGGER trg_update_running_total
7 AFTER INSERT ON sales
8 FOR EACH ROW
9 DECLARE
10     total_sales NUMBER;
11 BEGIN
12     -- Calculate the running total after the new row is inserted
13     SELECT NVL(SUM(amount), 0) + :NEW.amount
14     INTO total_sales
15     FROM sales;
16
17     -- Update the running total for the newly inserted row
18     UPDATE sales
19     SET running_total = total_sales
20     WHERE sale_id = :NEW.sale_id;
21 END;
22 /
23
24
```

The right-hand pane shows the execution output, which is empty, displaying "Program did not output anything!". The Windows taskbar at the bottom shows the time as 11:38 on 06-10-2024.