Ass3

Q1 Line 1
CREATE OR REPLACE TABLE emp (xxx);
CREATE OR REPLACE is not valid when creating tables.

CORRECTION : CREATE TABLE emp();

Q2 Line 8
CREATE OR REPLACE FUNCTION emp_stamp() RETURNS emp AS
A trigger function should return TRIGGER instead of the table.

CORRECTION : returns TRIGGER

Q3 Line 26
RETURN OLD;
We are updating data,there is no reason returning the old data...

CORRECTION : RETURN NEW;

Q4 Line 28
emp_stamp LANGUAGE sql;
LANGUAGE should be plpgsql instead of sql,because sql function cannot return a trigger.

CORRECTION : emp_stamp LANGUAGE plpgsql;

Q5 Line 30
CREATE OR REPLACE TRIGGER emp_stamp AFTER INSERT OR UPDATE ON emp
The main function is to check whether the to-be-inserted data is valid(whether empname and salary are given,salary value > 0 .etc), so the trigger must be executed before INSERT or UPDATE.

CORRECTION : CREATE OR REPLACE TRIGGER emp_stamp BEFORE INSERT OR UPDATE ON emp

full corrected code(see on the next page, corrected parts marked as blue):

```
CREATE TABLE emp
(
        empname      text,
        salary       integer,
        last_date timestamp,
        last_user text
);

CREATE OR REPLACE FUNCTION emp_stamp() RETURNS TRIGGER AS
$emp_stamp$
BEGIN
        -- Check that empname and salary are given
        IF NEW.empname IS NULL THEN
                RAISE EXCEPTION 'empname cannot be null';
        END IF;

        IF NEW.salary IS NULL THEN
                RAISE EXCEPTION '% cannot have null salary', NEW.empname;
        END IF;

        -- Invalid salary value
        IF NEW.salary < 0 THEN
                RAISE EXCEPTION '% cannot have a negative salary', NEW.empname;
        END IF;

        -- Record who changed the payroll and when
        NEW.last_date := current_timestamp;
        NEW.last_user := current_user;
        RETURN NEW;
END;
$emp_stamp$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER emp_stamp
        BEFORE INSERT OR UPDATE
        ON emp
        FOR EACH ROW
EXECUTE FUNCTION emp_stamp();
```