

MODEL ANSWERS :: Examples Clinic 5

Advanced SQL and Transactions

T1.

- I. **PL/SQL stands for Procedural Language SQL. It is a limited programming language that allows you to go beyond the relational database/SQL bounds by performing procedural (row-by-row) commands, unlike SQL which is a set-at-a-time declarative language. In other words, you can address sets of rows with SQL, but you do not ordinarily address individual rows with SQL. With PL/SQL, you can treat a table as a flat file that is accessed one row at a time.**
- II.


```
CREATE OR REPLACE PROCEDURE test1
    (firstname VARCHAR2, surname VARCHAR2, age NUMBER, salary NUMBER)
AS
BEGIN
    INSERT INTO staff VALUES (firstname, surname, age, salary);
END test1;
```
- III. **Triggers are mechanisms by which automatic reactive behaviour to events can be implemented in a database. Although reactive behaviour can be supported within applications, the addition of the semantics associated with event monitoring tasks to applications leads to the hiding, distribution and replication of this semantics among different application programs.**

There are four general uses for triggers:

1. **To enforce complex business rules (rules that are more complex than may be ordinarily available with a SQL constraint or command).**
2. **To compute a value that is based on other values or circumstances.**
3. **To "audit" an insert, delete, or update action (Oracle provides auditing features, but triggers are less formal approach to auditing).**
4. **To implement security measures.**

IV.

```
CREATE OR REPLACE TRIGGER numBookings2
AFTER INSERT ON booking
FOR EACH ROW
DECLARE
    numPresent INTEGER;
BEGIN
    SELECT count(*) INTO numPresent FROM numBookings
    WHERE trainnum = :new.trainnum AND traintdate = :new.traintdate;
    IF (numPresent = 0) THEN
        INSERT INTO numBookings VALUES (:new.trainnum, :new.traintdate, 1);
    ELSE
        UPDATE numBookings SET num = num + 1
        WHERE train# = :new.trainnum and traintdate = :new.traintdate;
    END IF;
END;
```

T2.

- I. **S:** $r_1(x) \quad w_2(z) \quad w_1(y)$
- II. **S:** $r_1(z) \quad r_2(q) \quad w_2(z) \quad r_1(q) \quad w_1(y)$
is equivalent to the T_1, T_2 serial schedule but not to the T_2, T_1 serial schedule because, with respect to the latter, the concurrent schedule could end with a different value for z , a variable that T_1 reads and T_2 writes.