Databases Autumn 2025

Hand-In Exercise 4

November 19, 2025

Aiysha Frutiger
Jannick Seper
Luis Tritschler

Total Points	

Task	Points

Databases
Autumn 2025
Exercise 4

Aiysha Frutiger
Jannick Seper
Luis Tritschler
Page 1 of 3

Task 1

a)

```
1  CREATE FUNCTION checkBalance (cid IN NUMBER)
2  RETURN VARCHAR2
3  IS
4          balanceCalculated NUMBER(10,2);
5          balanceSaved NUMBER(10,2);
6  BEGIN
7          SELECT SUM(amount)
8          INTO balanceCalculated
9          FROM BankTransaction
10         WHERE customer_id = cid;
11         SELECT balance
12         INTO balanceSaved
13         FROM Customer
14         WHERE customer_id = cid;
15
16         IF balanceCalculated = balanceSaved THEN
17                 RETURN 'Consistent';
18         ELSE
19                 RETURN 'Inconsistent';
20         END IF;
21 END;
```

b)

```
1  CREATE PROCEDURE checkAllBalances (consistent OUT VARCHAR2)
2  IS
3          inconsistent BOOLEAN := FALSE;
4          CURSOR c IS SELECT customer_id FROM Customer;
5  BEGIN
6          FOR record IN c LOOP
7                  IF check_balance(record.customer_id) =
8                                          'Inconsistent' THEN
9                          inconsistent := TRUE;
10                 END IF;
11         END LOOP;
12
13         IF inconsistent THEN
14                 consistent := 'Inconsistent';
15         ELSE
16                 consistent := 'Consistent';
17         END IF;
18 END;
```

Databases
Autumn 2025
Exercise 4

Aiysha Frutiger
Jannick Seper
Luis Tritschler
Page 2 of 3

c)

```
1   CREATE PROCEDURE check_receive (cid IN INT, t_amount IN DECIMAL(10,2),
2                        t_description IN VARCHAR(255))
3   IS
4           cust_exists INT;
5           not_exists_exception EXCEPTION;
6           max_tid INT;
7   BEGIN
8           SELECT COUNT(*)
9           INTO cust_exists
10          FROM Customer
11          WHERE customer_id = cid;
12
13          IF cust_exists = 0 THEN RAISE not_exists_exception;
14
15          UPDATE Customer
16          SET balance = balance + t_amount
17          WHERE customer_id = cid;
18
19          SELECT MAX(transaction_id)
20          INTO max_tid
21          FROM Transaction;
22
23          INSERT INTO Transaction
24          VALUES (max_tid + 1, cid, SYSDATE, t_amount, t_description);
25  EXCEPTION
26          WHEN not_exists_exception THEN
27                  ROLLBACK;
28          WHEN OTHERS THEN
29                  ROLLBACK;
30  END;
```

Databases
Autumn 2025
Exercise 4

Aiysha Frutiger
Jannick Seper
Luis Tritschler
Page 3 of 3

Task 2

Task 3

Task 4

Task 5