

23K-6005

HARIS AHMED

BSAI-5A

## DATABASE LAB 10:

### LAB TASKS

1a)

The screenshot shows the Oracle SQL Developer interface with a connection named 'my\_hr\_conn'. The 'Worksheet' tab contains a script for managing bank accounts. The script inserts three accounts (Account A, B, C) with initial balances of 10000, 5000, and 8000 respectively. It then performs a transaction: deducting 5000 from Account A, adding 5000 to Account B, and adding 1000 to Account C. The 'Script Output' tab shows the results of the execution, indicating 1 row updated for each step, and a final rollback.

```
INSERT INTO bank_accounts VALUES (1, 'Account A', 10000);
INSERT INTO bank_accounts VALUES (2, 'Account B', 5000);
INSERT INTO bank_accounts VALUES (3, 'Account C', 8000);

-- Commit the initial inserts (implicit or explicit depending on AUTOCOMMIT; assume OFF)
COMMIT;

-- Show initial balances
SELECT * FROM bank_accounts;

-- Start the transaction (implicitly starts with first DML)
-- 1. Deduct 5000 from Account A
UPDATE bank_accounts SET balance = balance - 5000 WHERE holder_name = 'Account A';

-- 2. Credit 5000 to Account B
UPDATE bank_accounts SET balance = balance + 5000 WHERE holder_name = 'Account B';

-- 3. Update balance of Account C by mistake (e.g., add 1000)
UPDATE bank_accounts SET balance = balance + 1000 WHERE holder_name = 'Account C';

-->>Query Run In:Query Result 5
1 row updated.

-->>Query Run In:Query Result 6
1 row updated.

-->>Query Run In:Query Result 7
1 row updated.

-->>Query Run In:Query Result 8
Rollback complete.
```

1b)

Oracle SQL Developer : my\_hr\_conn~1

File Edit View Navigate Run Source Team Tools Window Help

Connections

- my\_hr\_conn
- my\_hr\_conn (Filtered)
- Views
- Indexes
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

DBA

Connections

Click on an identifier with the Control key down to perform "Go to Declaration"

Line 36 Column 29 | Insert | Modified | Windows: 0

```
-- Show initial balances
SELECT * FROM bank_accounts;

-- Start the transaction (implicitly starts with first DML)
-- 1. Deduct 5000 from Account A
UPDATE bank_accounts SET balance = balance - 5000 WHERE holder_name = 'Account A';

-- 2. Credit 5000 to Account B
UPDATE bank_accounts SET balance = balance + 5000 WHERE holder_name = 'Account B';

-- 3. Update balance of Account C by mistake (e.g., add 1000)
UPDATE bank_accounts SET balance = balance + 1000 WHERE holder_name = 'Account C';

-- Show changes before commit (temporarily)
SELECT * FROM bank_accounts;

-- 4. Use ROLLBACK to undo all changes
ROLLBACK;

-- 5. Show that balances returned to original values
SELECT * FROM bank_accounts;
```

Script Output X | Query Result 6 X | Query Result 7 X

All Rows Fetched: 3 in 0.001 seconds

ACCOUNT_NO	HOLDER_NAME	BALANCE
1	Account A	10000
2	Account B	5000
3	Account C	8000

2a)

Oracle SQL Developer : my\_hr\_conn~1

File Edit View Navigate Run Source Team Tools Window Help

Connections

- my\_hr\_conn
- my\_hr\_conn (Filtered)
- Views
- Indexes
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

DBA

Connections

Click on an identifier with the Control key down to perform "Go to Declaration"

Line 21 Column 1 | Insert | Modified | Windows: 0

```
quantity NUMBER
);

INSERT INTO inventory VALUES (1, 'Item 1', 100);
INSERT INTO inventory VALUES (2, 'Item 2', 200);
INSERT INTO inventory VALUES (3, 'Item 3', 300);
INSERT INTO inventory VALUES (4, 'Item 4', 400);

COMMIT;
```

Script Output X | Query Result 1 X | Task completed in 0.43 seconds

Table INVENTORY created.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Commit complete.

>>Query Run In:Query Result 2

1 row updated.

Savepoint created.

1 row updated.

Savepoint created.

2b)

Oracle SQL Developer : my\_hr\_conn~1

```

-- Create a new table named 'inventory' with columns item_id, item_name, and quantity.
CREATE TABLE inventory (
    item_id NUMBER PRIMARY KEY,
    item_name VARCHAR2(50),
    quantity NUMBER
);

-- Insert initial data into the 'inventory' table.
INSERT INTO inventory VALUES (1, 'Item 1', 100);
INSERT INTO inventory VALUES (2, 'Item 2', 200);
INSERT INTO inventory VALUES (3, 'Item 3', 300);
INSERT INTO inventory VALUES (4, 'Item 4', 400);

-- Commit the transaction.
COMMIT;

-- Select all data from the 'inventory' table.
SELECT * FROM inventory;

-- Update the 'quantity' column for item_id 1 by subtracting 10.
UPDATE inventory SET quantity = quantity - 10 WHERE item_id = 1;

-- Savepoint 'sp1'.
SAVEPOINT sp1;

-- Update the 'quantity' column for item_id 2 by subtracting 20.
UPDATE inventory SET quantity = quantity - 20 WHERE item_id = 2;

-- Savepoint 'sp2'.
SAVEPOINT sp2;

```

Script Output | Query Result 1 | All Rows Fetched: 4 in 0.002 seconds

ITEM_ID	ITEM_NAME	QUANTITY
1	Item 1	90
2	Item 2	200
3	Item 3	300
4	Item 4	400

3a)

Oracle SQL Developer : my\_hr\_conn~1

```

-- Show initial records
SELECT * FROM fees;

-- Start the transaction
-- 1. Update amount_paid for Student 1 (e.g., add 5000)
UPDATE fees SET amount_paid = amount_paid + 5000 WHERE student_id = 1;

-- 2. Savepoint halfway
SAVEPOINT halfway;

-- 3. Update amount_paid for Student 2 (e.g., add 6000)
UPDATE fees SET amount_paid = amount_paid + 6000 WHERE student_id = 2;

-- 4. Due to error, rollback to halfway (undo Student 2 update)
ROLLBACK TO halfway;

-- 5. Commit the correct part (only Student 1 is updated)
COMMIT;

-- Show final records
SELECT * FROM fees;

```

Script Output | Query Result 6 | Task completed in 0.457 seconds

```

>>>Query Run InQuery Result 8

1 row updated.

Savepoint created.

1 row updated.

Rollback complete.

Commit complete.

```

3b)

Oracle SQL Developer : my\_hr\_conn~1

```
-- Show initial records
SELECT * FROM fees;

-- Start the transaction
-- 1. Update amount_paid for Student 1 (e.g., add 5000)
UPDATE fees SET amount_paid = amount_paid + 5000 WHERE student_id = 1;

-- 2. Savepoint halfway
SAVERPOINT halfway;

-- 3. Update amount_paid for Student 2 (e.g., add 4000)
UPDATE fees SET amount_paid = amount_paid + 4000 WHERE student_id = 2;

-- 4. Due to error = rollback to halfway (undo Student 2 update)
ROLLBACK TO halfway;

-- 5. Commit the correct part (only Student 1 is updated)
COMMIT;

-- Show final records
SELECT * FROM fees;
```

Script Output | Query Result 6 | Query Result 7 | Query Result 8 | Query Result 9 |

STUDENT_ID	NAME	AMOUNT_PAID	TOTAL_FEE
1	Student 1	5000	10000
2	Student 2	0	10000
3	Student 3	0	10000

4a)

Oracle SQL Developer : my\_hr\_conn~1

```
DROP TABLE orders;
DROP TABLE products;

CREATE TABLE products (
    product_id NUMBER PRIMARY KEY,
    product_name VARCHAR2(50),
    stock NUMBER
);

CREATE TABLE orders (
    order_id NUMBER PRIMARY KEY,
    product_id NUMBER,
    quantity NUMBER
);

INSERT INTO products VALUES (1, 'Product 1', 50);
INSERT INTO products VALUES (2, 'Product 2', 60);
INSERT INTO orders VALUES (1, 1, 10);

COMMIT;

SELECT * FROM products;
SELECT * FROM orders;

UPDATE products SET stock = stock - 5 WHERE product_id = 1;
```

Script Output | Query Result | Query Result 1 |

ORDER_ID	CUSTOMER_ID
1	1
2	2

4b)

Oracle SQL Developer : my\_hr\_conn~1

```

SELECT * FROM products;
SELECT * FROM orders;

UPDATE products SET stock = stock - 5 WHERE product_id = 1;

INSERT INTO orders (order_id, product_id, quantity) VALUES (2, 1, 5);

DELETE FROM products WHERE product_id = 2;

SELECT * FROM products;
SELECT * FROM orders;

ROLLBACK;

SELECT * FROM products;
SELECT * FROM orders;

UPDATE products SET stock = stock - 5 WHERE product_id = 1;
INSERT INTO orders (order_id, product_id, quantity) VALUES (2, 1, 5);

COMMIT;

SELECT * FROM products;
SELECT * FROM orders;

```

Script Output | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5 | Query Result 6 | Query Result 7

PRODUCT_ID	PRODUCT_NAME	STOCK
1	Product 1	50
2	Product 2	60

4c)

Oracle SQL Developer : my\_hr\_conn~1

```

SELECT * FROM products;
SELECT * FROM orders;

UPDATE products SET stock = stock - 5 WHERE product_id = 1;

INSERT INTO orders (order_id, product_id, quantity) VALUES (2, 1, 5);

DELETE FROM products WHERE product_id = 2;

SELECT * FROM products;
SELECT * FROM orders;

ROLLBACK;

SELECT * FROM products;
SELECT * FROM orders;

UPDATE products SET stock = stock - 5 WHERE product_id = 1;
INSERT INTO orders (order_id, product_id, quantity) VALUES (2, 1, 5);

COMMIT;

SELECT * FROM products;
SELECT * FROM orders;

```

Script Output | Query Result 1 | Query Result 2 | Query Result 3 | Query Result 4 | Query Result 5 | Query Result 6 | Query Result 7

PRODUCT_ID	PRODUCT_NAME	STOCK
1	Product 1	45
2	Product 2	60

## 5a and b)

Oracle SQL Developer : my\_hr\_conn~1

```
-- 2. SAVEPOINT A
SAVEPOINT A;

-- 3. Increase salary of emp 2 (e.g., by 1000)
UPDATE employees SET salary = salary + 1000 WHERE emp_id = 2;

-- 4. SAVEPOINT B
SAVEPOINT B;

-- 5. Increase salary of emp 3 (e.g., by 1000)
UPDATE employees SET salary = salary + 1000 WHERE emp_id = 3;

-- 6. Rollback to SAVEPOINT A (undo changes to emp 2 and 3)
ROLLBACK TO A;

-- 7. Commit
COMMIT;

-- Show final salaries
SELECT * FROM employees;
```

Script Output: | Query Result 6 | Query Result 7 | Query Result 8 | Query Result 9 | Query Result 10 | Query Result 11 | Query Result 12 | Query Result 13 | Query Result 14 | Query Result 15 | Query Result 16 | Query Result 17 | Query Result 18 | Task completed in 0.586 seconds

DBA | Connections

All Reports | Analytic View Reports | Data Dictionary Reports | OLAP Reports | TimesTen Reports | User Defined Reports

Click on an identifier with the Control key down to perform "Go to Declaration"

Oracle SQL Developer : my\_hr\_conn~1

```
-- 2. SAVEPOINT A
SAVEPOINT A;

-- 3. Increase salary of emp 2 (e.g., by 1000)
UPDATE employees SET salary = salary + 1000 WHERE emp_id = 2;

-- 4. SAVEPOINT B
SAVEPOINT B;

-- 5. Increase salary of emp 3 (e.g., by 1000)
UPDATE employees SET salary = salary + 1000 WHERE emp_id = 3;

-- 6. Rollback to SAVEPOINT A (undo changes to emp 2 and 3)
ROLLBACK TO A;

-- 7. Commit
COMMIT;

-- Show final salaries
SELECT * FROM employees;
```

Script Output: | Query Result 6 | Query Result 7 | Query Result 8 | Query Result 9 | Query Result 10 | Query Result 11 | Query Result 12 | Query Result 13 | Query Result 14 | Query Result 15 | Query Result 16 | Query Result 17 | Query Result 18 | All Rows Fetched: 6 in 0.003 seconds

EMPLOYEE_ID	EMPLOYEE_NAME	SALARY
1	2 Emp 2	6000
2	10 Emp 4	75000
3	1 Emp 1	5000
4	3 Emp 3	7000
5	4 Emp 4	8000
6	5 Emp 5	9000

DBA | Connections

All Reports | Analytic View Reports | Data Dictionary Reports | OLAP Reports | TimesTen Reports | User Defined Reports

Click on an identifier with the Control key down to perform "Go to Declaration"