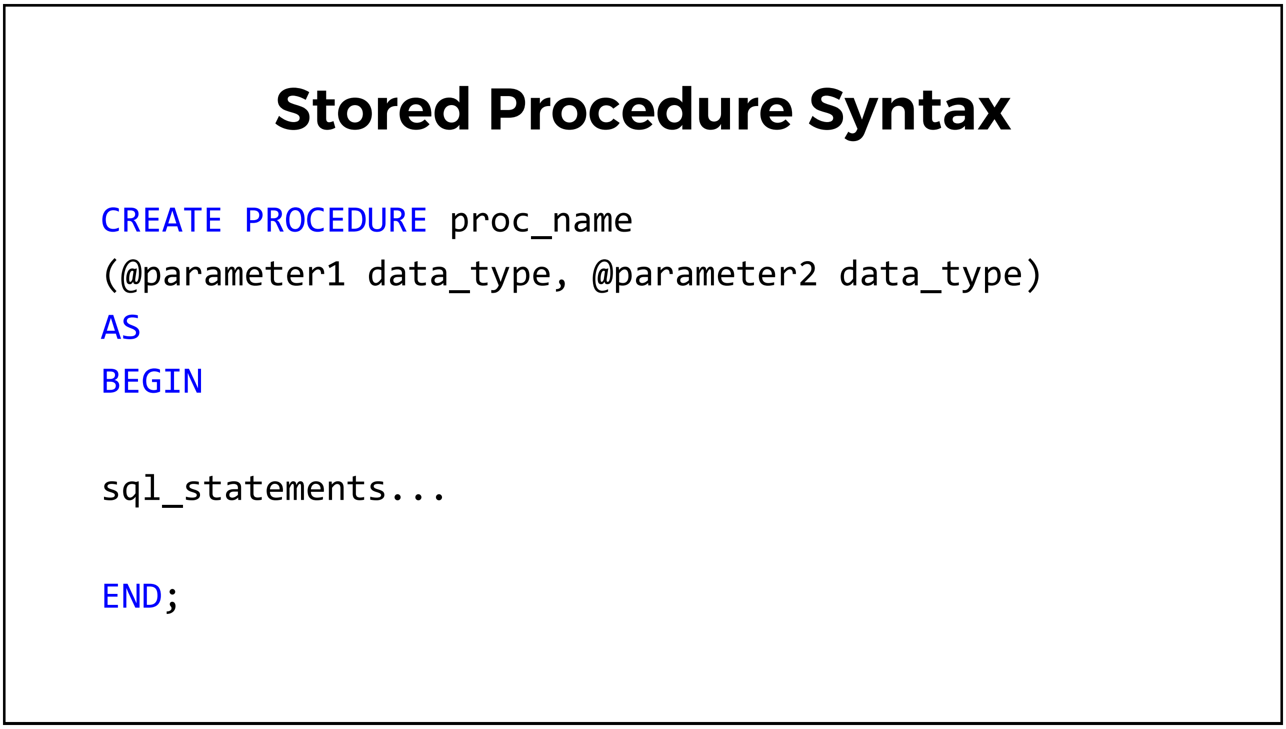
Stored Procedure Challenges

1



Stored Procedure Syntax

```
CREATE PROCEDURE proc_name
(@parameter1 data_type, @parameter2 data_type)
AS
BEGIN

sql_statements...

END;
```

2

Note!

- Make sure you have created the oes.bank_accounts and oes.bank_transactions tables before doing the third challenge.
- To create the tables, execute the SQL file attached as a resource to this lecture.

Challenge 1

Create a stored procedure called `oes.getQuantityOnHand` that returns the `quantity_on_hand` in the `oes.inventories` table for a given `product_id` and `warehouse_id`.

Execute the stored procedure to return the quantity on hand of product id 4 at warehouse id 2.

Hints:

Define two input parameters i.e., `@product_id` and `@warehouse_id` of data type `INT`. Also, reference these parameters in the `WHERE` clause of the `SELECT` statement.

Challenge 2

Create a stored procedure called oes.getCurrentProducts that returns current products (discontinued = 0) in the oes.products table. In addition, define two input parameters:

- A parameter called @product_name of data type VARCHAR(100). Allow users to wildcard search on the product_name.
- A parameter called @max_list_price of data type DECIMAL(19,4). Allow users to only include current products that have a list_price that is less than or equal to a specified value for this parameter.

Execute the stored procedure to return current products that contain the word 'Drone' and have a maximum price of $700.

5

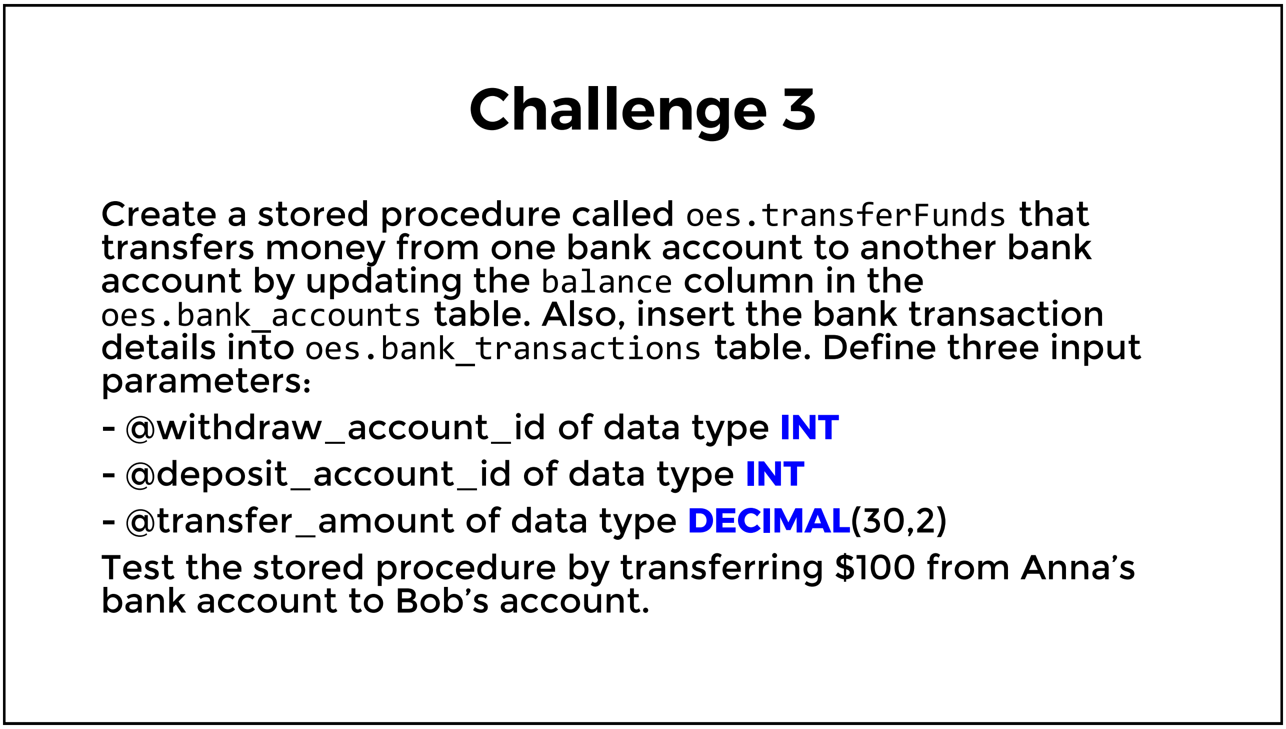
Hints for Challenge 2

Recall that we can allow for wildcard searches by using the LIKE operator with '%' concatenated on to one or both sides of the input parameter. For example:

```
WHERE first_name LIKE '%' + @name + '%'
```

Also include another condition in the WHERE clause to check that list_price is less than or equal to the @max_list_price parameter.

6



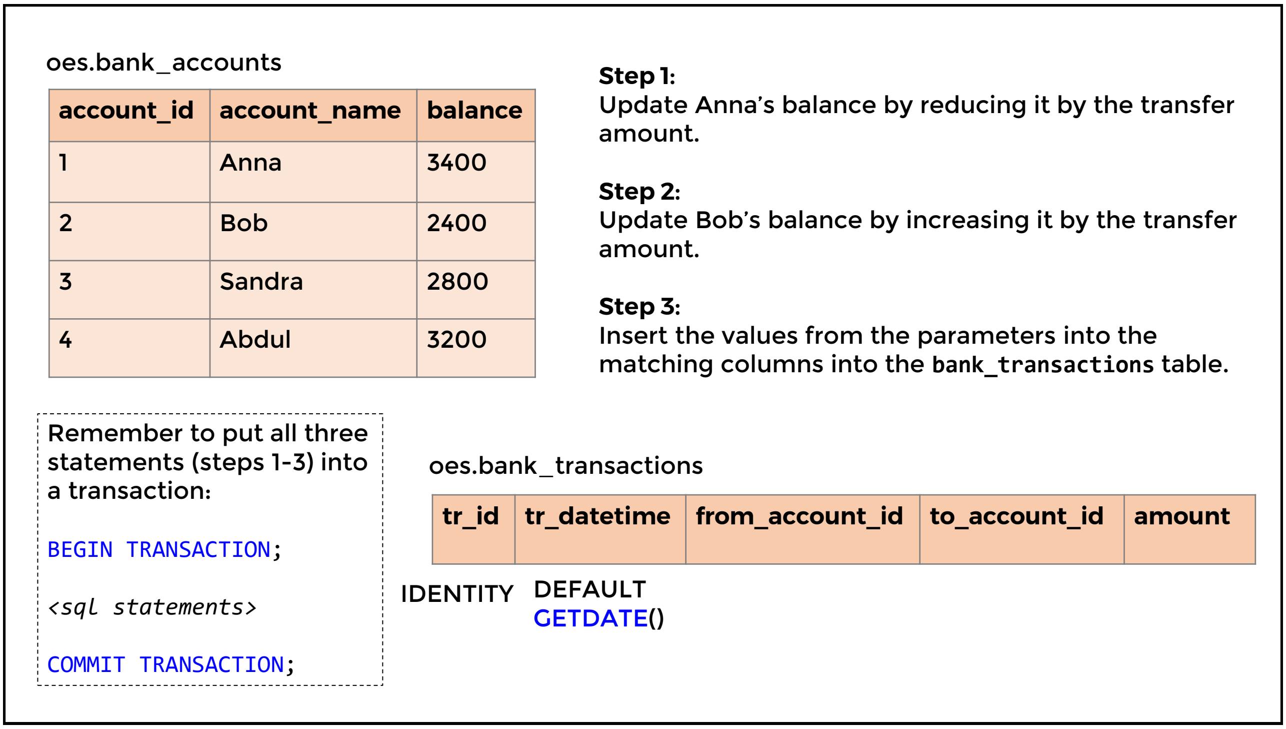
Challenge 3

Create a stored procedure called `oes.transferFunds` that transfers money from one bank account to another bank account by updating the `balance` column in the `oes.bank_accounts` table. Also, insert the bank transaction details into `oes.bank_transactions` table. Define three input parameters:

- @withdraw_account_id of data type **INT**
- @deposit_account_id of data type **INT**
- @transfer_amount of data type **DECIMAL**(30,2)

Test the stored procedure by transferring $100 from Anna’s bank account to Bob’s account.

7



oes.bank_accounts

account_id	account_name	balance
1	Anna	3400
2	Bob	2400
3	Sandra	2800
4	Abdul	3200

Step 1:
Update Anna’s balance by reducing it by the transfer amount.

Step 2:
Update Bob’s balance by increasing it by the transfer amount.

Step 3:
Insert the values from the parameters into the matching columns into the `bank_transactions` table.

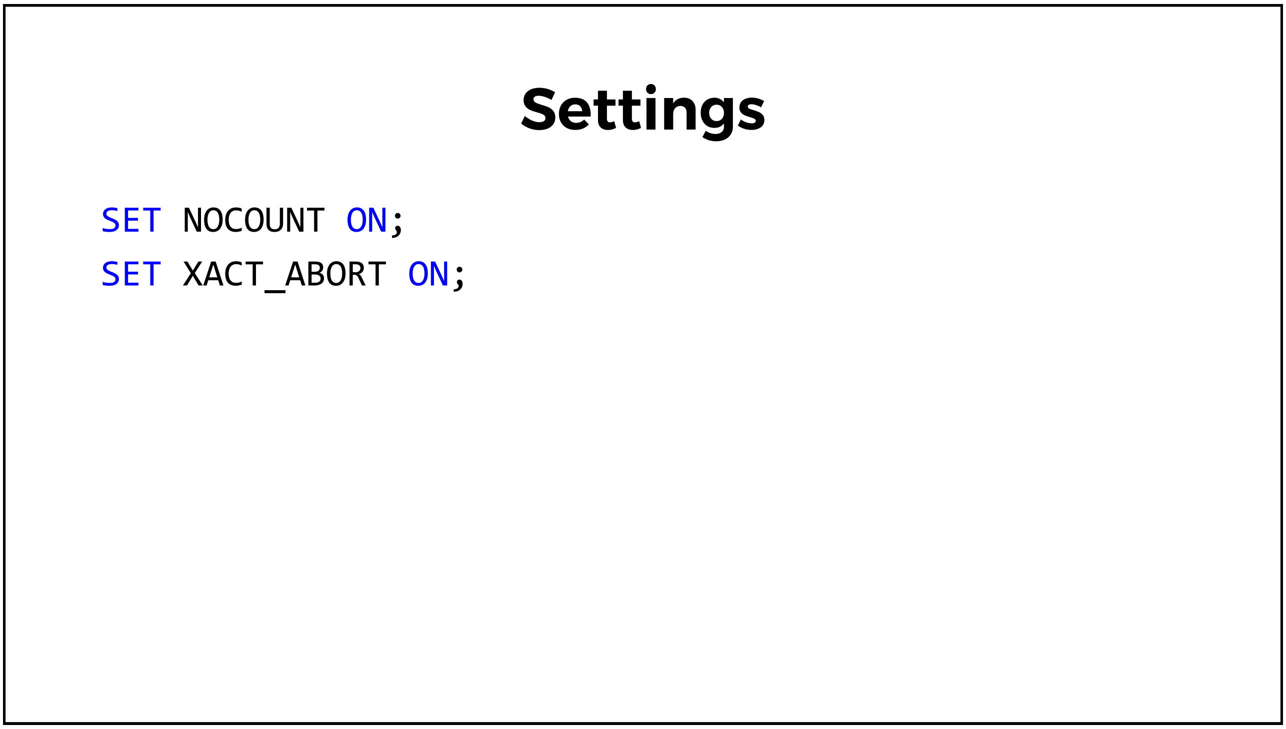
Remember to put all three statements (steps 1-3) into a transaction:

```
BEGIN TRANSACTION;

<sql statements>

COMMIT TRANSACTION;
```

oes.bank_transactions

tr_id	tr_datetime	from_account_id	to_account_id	amount
IDENTITY	DEFAULT GETDATE()			

8



Settings

```
SET NOCOUNT ON;
SET XACT_ABORT ON;
```

9