ORACLE

Trigger

Mag. Thomas Griesmayer

Local check

200

€ -150 ?

```
SELECT AMOUNT
FROM   CUSTOMER
WHERE  ACCOUNT_ID = 3;
```

CUSTOMER		
CUSTOMER_ID	NAME	AMOUNT
1	Thomas	0
2	Andrea	600
3	Verena	200
4	Marion	100

Local check

200

€ -150

```
SELECT AMOUNT
FROM   CUSTOMER
WHERE  ACCOUNT_ID = 3;
```

200

€ -110?

CUSTOMER		
CUSTOMER_ID	NAME	AMOUNT
1	Thomas	0
2	Andrea	600
3	Verena	200
4	Marion	100

Local check

200

€ -150

```
UPDATE FROM CUSTOMER
SET    AMOUNT = AMOUNT-150
WHERE  ACCOUNT_ID = 3;
```

CUSTOMER		
CUSTOMER_ID	NAME	AMOUNT
1	Thomas	0
2	Andrea	600
3	Verena	50
4	Marion	100

200

€ -110

Local check

200

```
UPDATE FROM CUSTOMER
SET    AMOUNT = AMOUNT-110
WHERE  ACCOUNT_ID = 3;
```

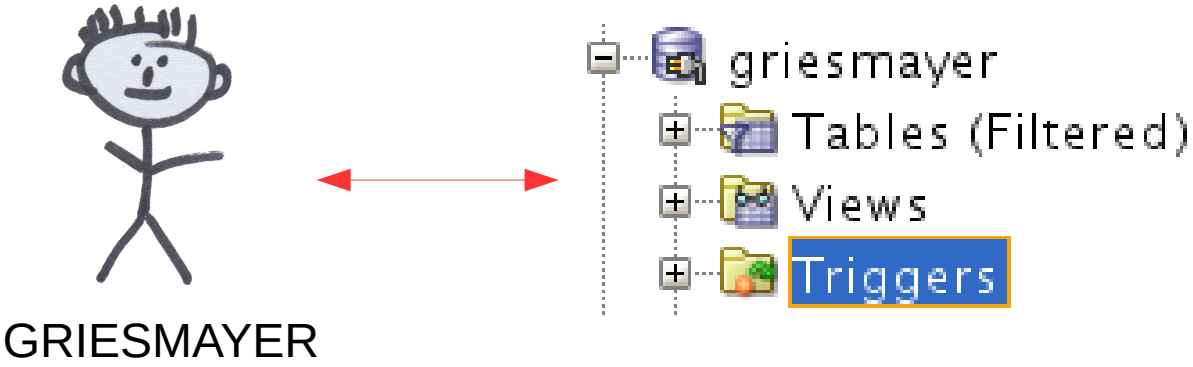
200

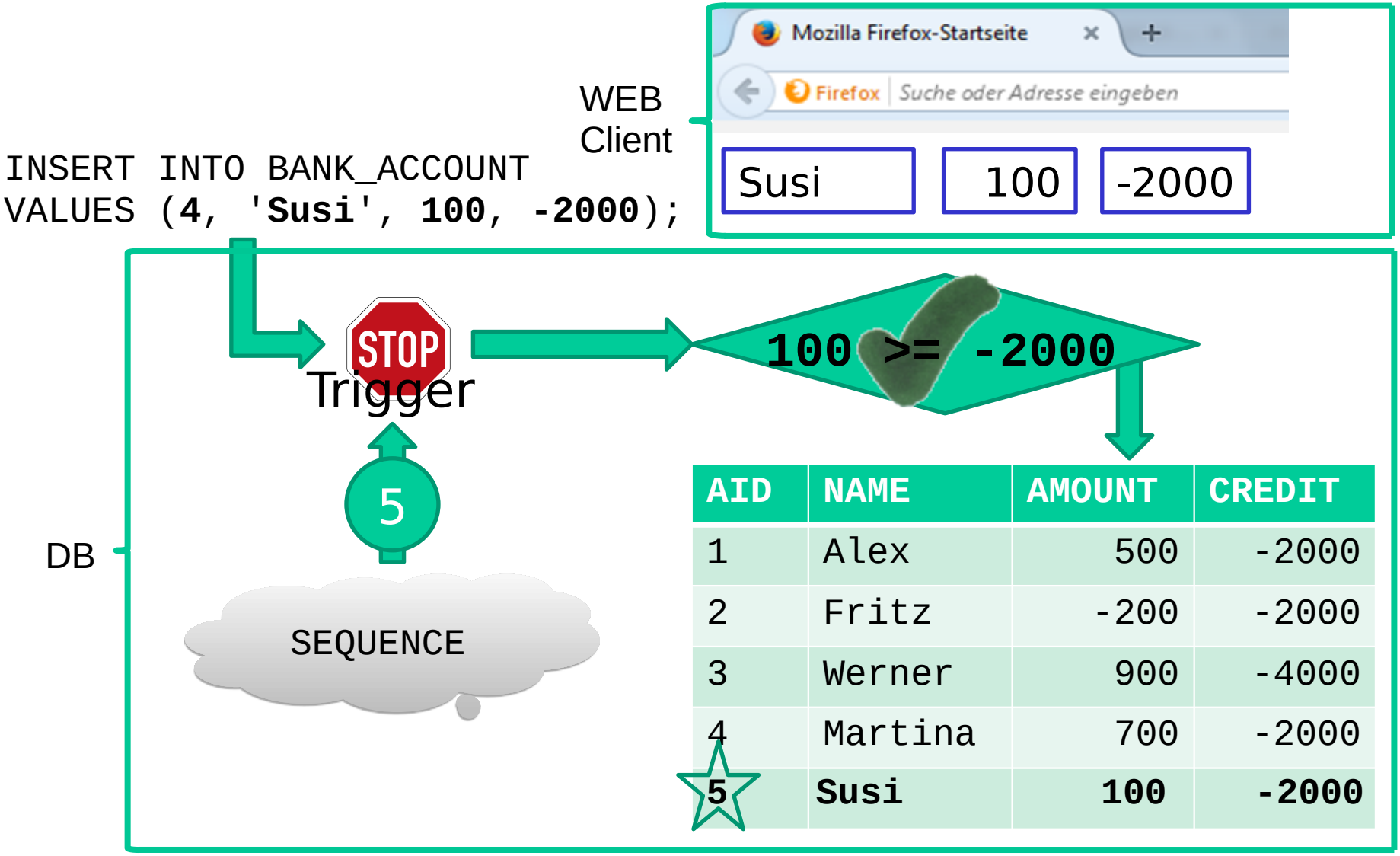
€ -110

CUSTOMER		
CUSTOMER_ID	NAME	AMOUNT
1	Thomas	0
2	Andrea	600
3	Verena	-60
4	Marion	100

Trigger

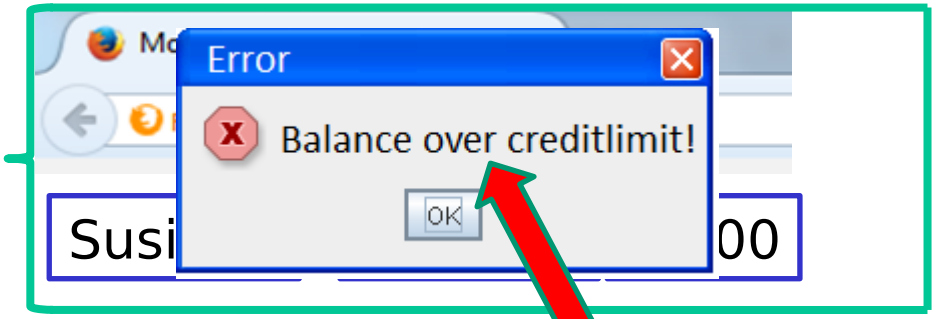
- Use the `CREATE TRIGGER` statement to create a trigger, that is stored in the database.
- Triggers executes a `PL/SQL` block when an `INSERT`, `UPDATE`, or `DELETE` occurred on a particular table or view.

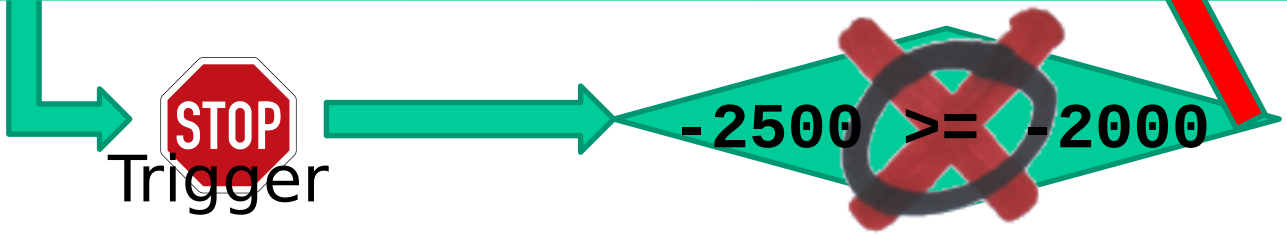



WEB
Client
Mozilla Firefox-Startseite
Firefox
Suche oder Adresse eingeben
Susi
100
-2000
INSERT INTO BANK_ACCOUNT
VALUES (4, 'Susi', 100, -2000);
DB
STOP
Trigger
100 >= -2000
5
SEQUENCE
AID
NAME
AMOUNT
CREDIT
1
Alex
500
-2000
2
Fritz
-200
-2000
3
Werner
900
-4000
4
Martina
700
-2000
5
Susi
100
-2000

```
UPDATE FROM BANK_ACCOUNT
SET    AMOUNT = AMOUNT-2600
WHERE  ACCOUNT_ID = 5;
```

WEB Client





DB

AID	NAME	AMOUNT	CREDIT
1	Alex	500	-2000
2	Fritz	-200	-2000
3	Werner	900	-4000
4	Martina	700	-2000
5	Susi	100	-2000

Trigger timing

- A **BEFORE** row trigger executes the PL/SQL code before the row data is written.
- An **AFTER** row trigger writes the data first and then executes the PL/SQL code.

```
INSERT INTO BANK_ACCOUNT
VALUES (6, 'Max', 500, -2000);
```

```
CREATE TRIGGER TRG_ACCOUNT
BEFORE INSERT OR UPDATE
```

RID	AID	Name	AMOUNT	CREDIT
B8	6	Max	500	-2000

```
CREATE TRIGGER TRG_ACCOUNT
AFTER INSERT OR UPDATE
```

RID	AID	Name	AMOUNT	CREDIT
B8	6	Max	500	-2000

ROW content

```
UPDATE BANK_ACCOUNT
SET    AMOUNT = 400,
       CREDIT = -4000
WHERE  ACCOUNT_ID = 2;
```

2	Fritz	**400**	**-4000**

:new.CREDIT_LIMIT

:old.NAME

AID	NAME	AMOUNT	CREDIT
1	Alex	500	-2000
2	Fritz	**-200**	**-2000**
3	Werner	900	-4000

```
INSERT INTO BANK_ACCOUNT
VALUES (-17, 'Andrea', 900, -4000, '1911-01-01');
```

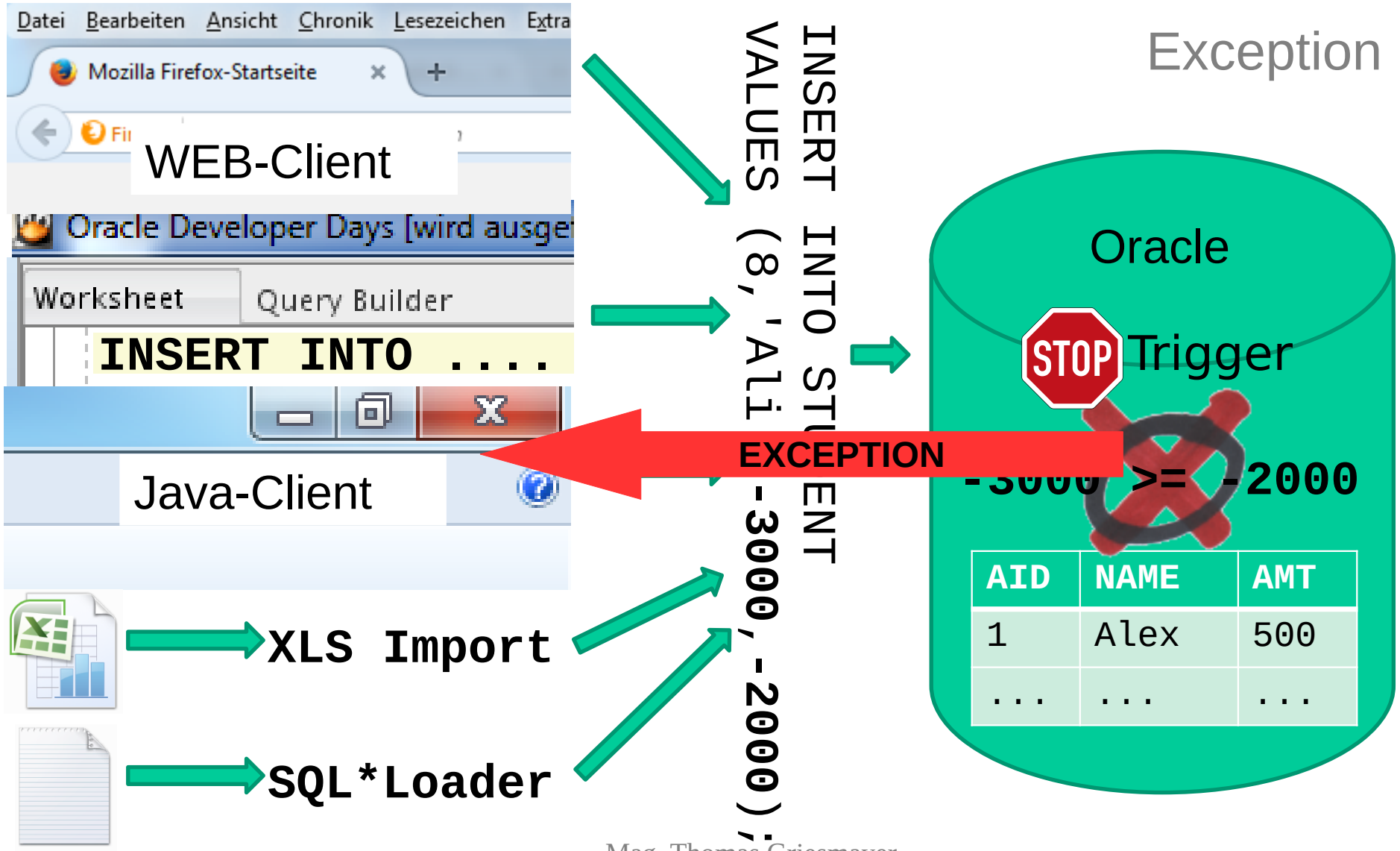
Trigger

```
SELECT SYSDATE INTO :new.LAST_TRANSACTION
FROM DUAL;
```

```
SELECT SEQ_ACCOUNT_ID.nextval INTO :new.ACCOUNT_ID
FROM DUAL;
```

RID	AID	Name	AMOUNT	CREDIT	LAST_TRX
B8	6	Max	500	-2000	2016-02-15
B9	~~-17~~ 7	Andrea	900	-4000	~~1911-01-01~~ 2018-11-05

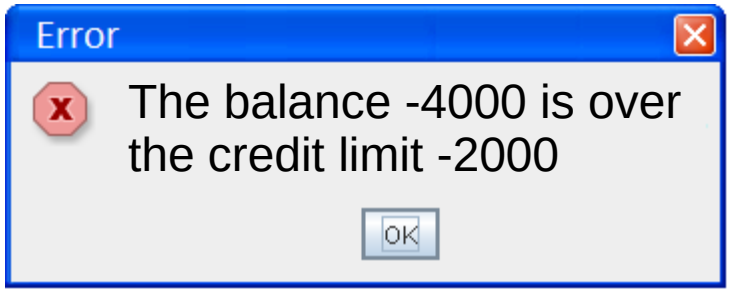
Exception


Datei Bearbeiten Ansicht Chronik Lesezeichen Extra
Mozilla Firefox-Startseite
WEB-Client
Oracle Developer Days [wird ausge
Worksheet
Query Builder
INSERT INTO ....
Java-Client
XLS Import
SQL*Loader
INSERT INTO STUDENT
VALUES (8,'Ali
-3000,-2000);
Oracle
STOP
Trigger
EXCEPTION
-3000 >= -2000
AID
NAME
AMT
1
Alex
500
...
...
...

Exception

AID	NAME	AMOUNT	CREDIT
1	Alex	500	-2000
2	Fritz	-200	-2000
3	Werner	900	-4000

```
UPDATE BANK_ACCOUNT
SET    AMOUNT = -4000
WHERE  ACCOUNT_ID = 2;
```




```
IF (:new.AMOUNT < :new.CREDIT) THEN
   ROLLBACK;
   raise_application_error (-20999,
         'The balance' || :new.BALANCE ||
         ' is over the credit limit ' || :new.CREDIT);
END IF;
```

Errorcode

Errormessage