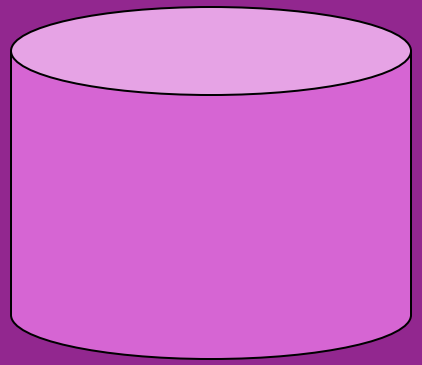


LEEDS BECKETT UNIVERSITY

School of Built Environment, Engineering and Computing

PL/SQL: Triggers

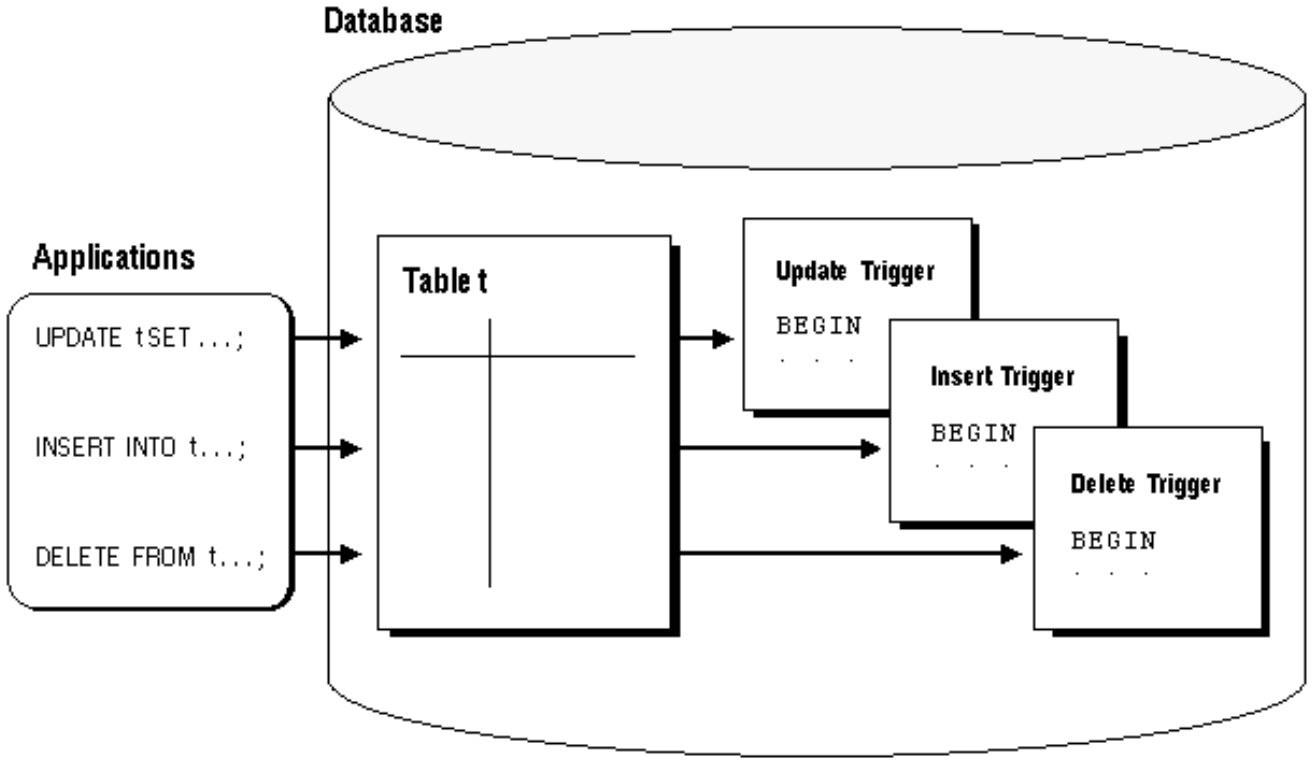
Sanela Lazarevski (Module Leader)

What is a Trigger?

- A PL/SQL procedure that is automatically fired (executed) when an INSERT, UPDATE or DELETE statement is issued against a table (not a view).
- Triggers stored in database (Server side)
- Can also be called from within APEX Form
- Similar to stored procedures but implicitly fired
- Procedure has to be specifically invoked

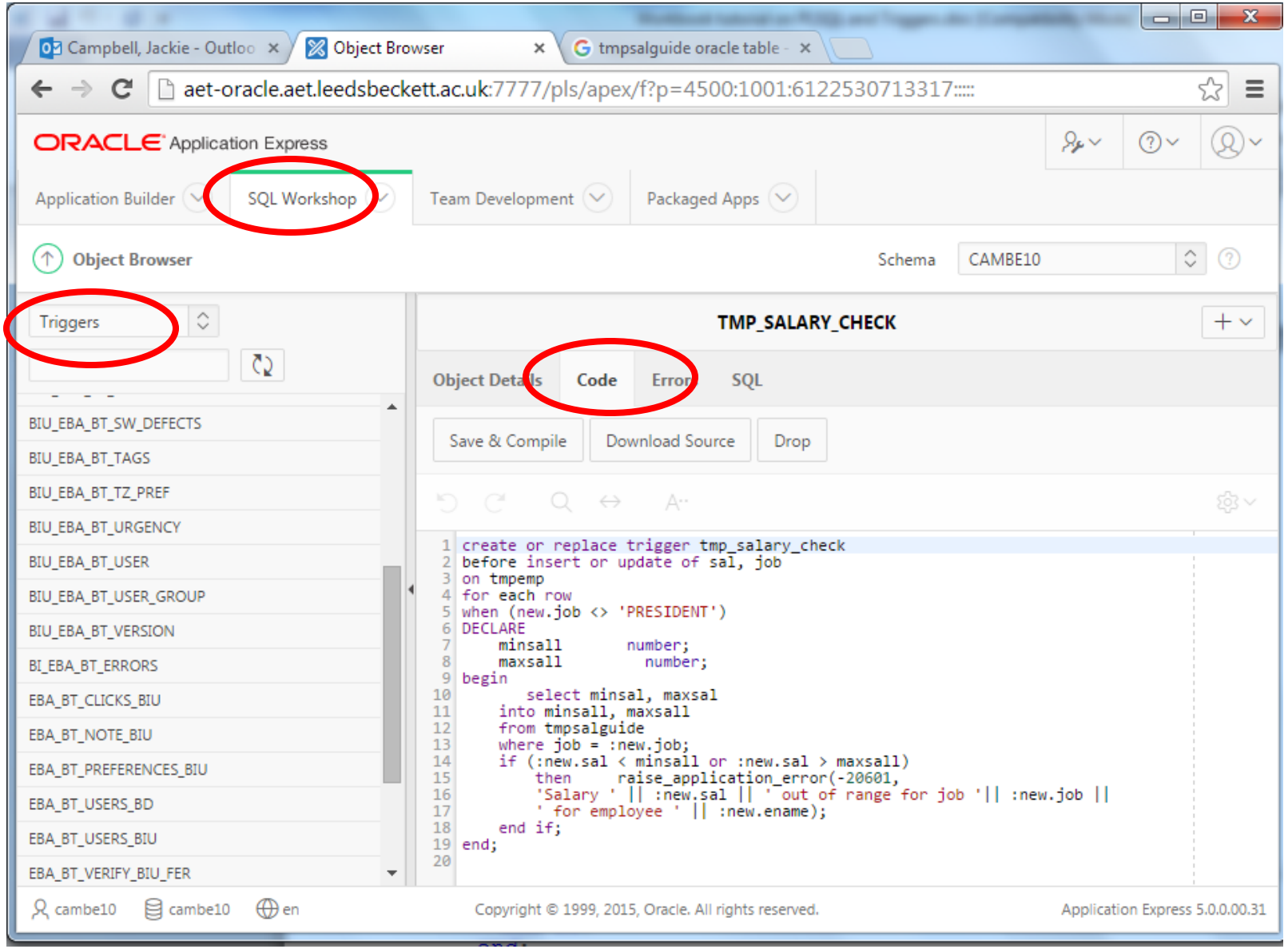
LEEDS BECKETT UNIVERSITY

Database Triggers



http://docs.oracle.com

Adding a Trigger using SQL Workshop



What are they used for?

- automatically generate derived column values
- prevent invalid transactions
- enforce complex security authorizations
- enforce referential integrity across nodes in a distributed database
- enforce complex business rules
- provide transparent event logging
- provide sophisticated auditing
- maintain synchronous table replicates
- gather statistics on table access

LEEDS BECKETT UNIVERSITY

http://docs.oracle.com

What is a Trigger?

- Triggered procedures usually just called triggers.
- A trigger is a piece of code that is automatically run when a specified event occurs.
- We say that the event causes the trigger to fire.

- Two types of Trigger
  - Row level
    - Fired each time table is affected by triggering statement
  - Statement level
    - Fired once by the statement irrespective of number of rows affected.

LEEDS BECKETT UNIVERSITY

Server side Triggers

- **There are three database events:**
  - **INSERT**
  - **UPDATE**
  - **DELETE**
- **A trigger may fire**
  - **BEFORE**
  - **or**
  - **AFTER** **a database event**
- **Oracle triggers can be**
  - **Row-level** **or** **Statement-level**

This gives us twelve different triggers:

BEFORE INSERT *row*
BEFORE INSERT *statement*
AFTER INSERT *row*
AFTER INSERT *statement*
BEFORE UPDATE *row* …

LEEDS BECKETT UNIVERSITY

Row-level Triggers

- Fire once for each table row affected by the transaction.
- Have access to both the old data and the new data in the current row.
- Often used to write audit information

  E.g. who inserted the row and when?

Statement-level Triggers

- Fire once for each transaction.
- Mainly used to enforce specific security measures on a table.

  E.g. Could allow accounts clerks access to the employee table at the financial year end, but not at other times.

Trigger Syntax

CREATE OR REPLACE TRIGGER *trigger_name*

[BEFORE | AFTER] [DELETE | INSERT | UPDATE]

ON *table_name*
[FOR EACH ROW]
[WHEN *condition*]

[*pl/sql block*];

Specifying the Triggered Action

```
CREATE OR REPLACE TRIGGER student_aft_ins_upd
AFTER INSERT OR UPDATE
ON student

WHEN USER = 'Stephen'
BEGIN
INSERT INTO user_tracking     (user_name, table, date)
          VALUES (USER, 'student', SYSDATE);
END;
```

The :old and :new Keywords

Row level trigger

```
BEGIN
  IF :old.fee_paid < :new.fee_paid
THEN
  INSERT INTO student_audit (user, date, row, action)
  VALUES (USER, SYSDATE,:old.student_id, 'fee payment');
ELSE
  …etc.
```

INSERTING, UPDATING & DELETING

AFTER trigger

```
IF INSERTING THEN
    INSERT INTO student_audit (user, date, row, action)
  VALUES (USER, SYSDATE, :old.student_id, 'insert');

ELSE IF DELETING THEN

    INSERT INTO student_audit (user, date, row, action)
  VALUES(USER, SYSDATE, :old.student_id, 'delete');
END IF
…
Etc.
```

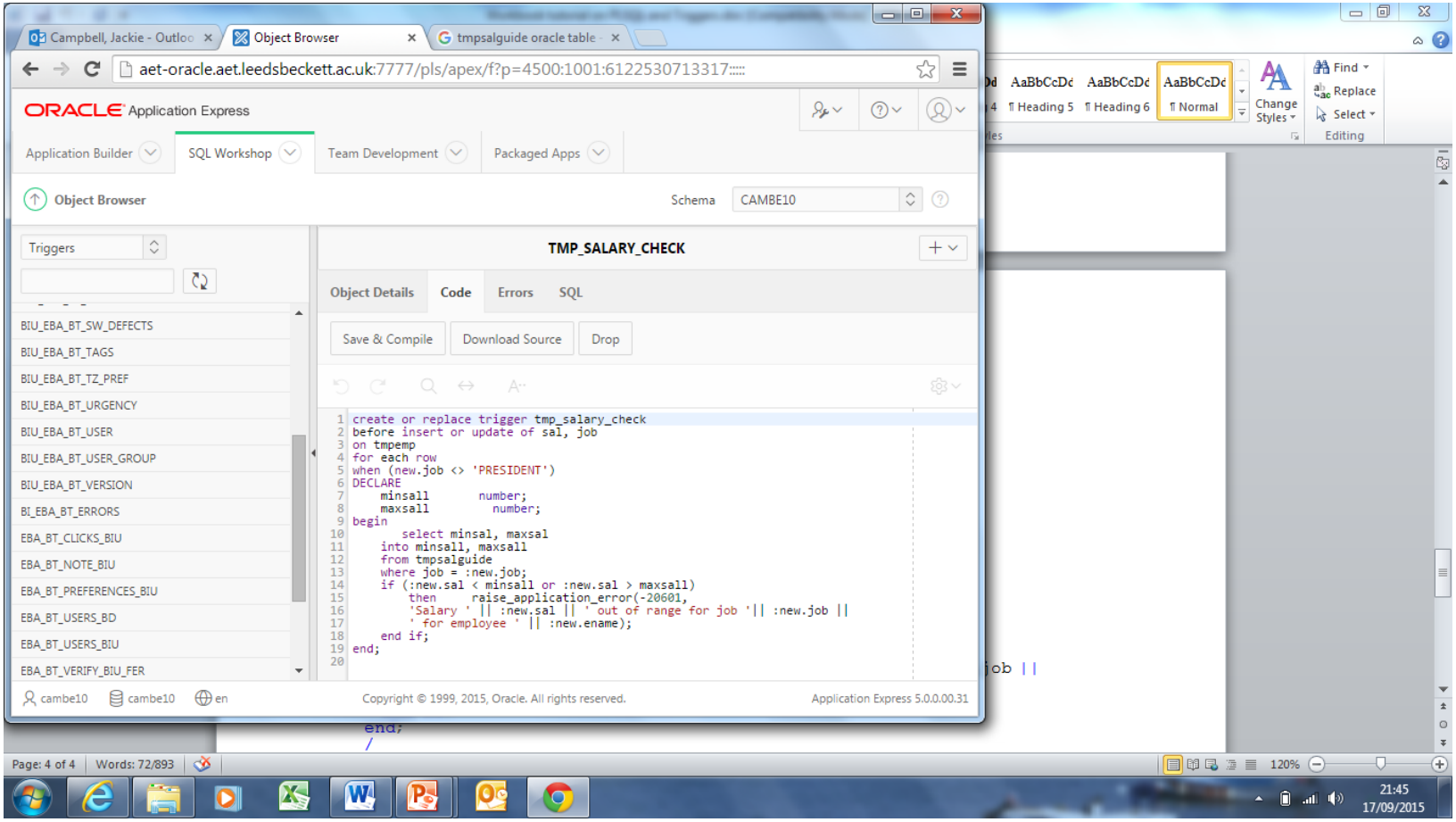
Exclusive arc type Trigger

```
create or replace trigger tmp_excl_arc
Before insert or update of cocustno, indcustno
on tmpord
for each row
begin
      if (:new.cocustno is null) and (:new.indcustno is null)
      then    raise_application_error(-20001,
              'this order must belong to a customer');
      end if;
      if (:new.cocustno is not null) and (:new.indcustno is not null)
      then    raise_application_error(-20002,
              'this order must belong to only one type of customer');
      end if;
end;
```

Statement level trigger – a weekend restriction

```
create or replace trigger tmp_permit_changes
before delete or insert or update
on tmpemp
declare
  dummy  integer;
begin
  /* if today is a sat or sun, then return an error */
  if (to_char(sysdate,'DY') = 'SAT' OR to_char(sysdate,'DY') = 'SUN')
  then raise_application_error(-20501,
    'May not change employee table during ' || to_char(sysdate,'DY'));
  end if;
end;
```

PL/SQL – trigger demo



Summary

- Why we need PL/SQL
- Structure
- Syntax and constructs
- Server Side Triggers

Thank you

Any questions?