

Christian D. Rosado Arroyo

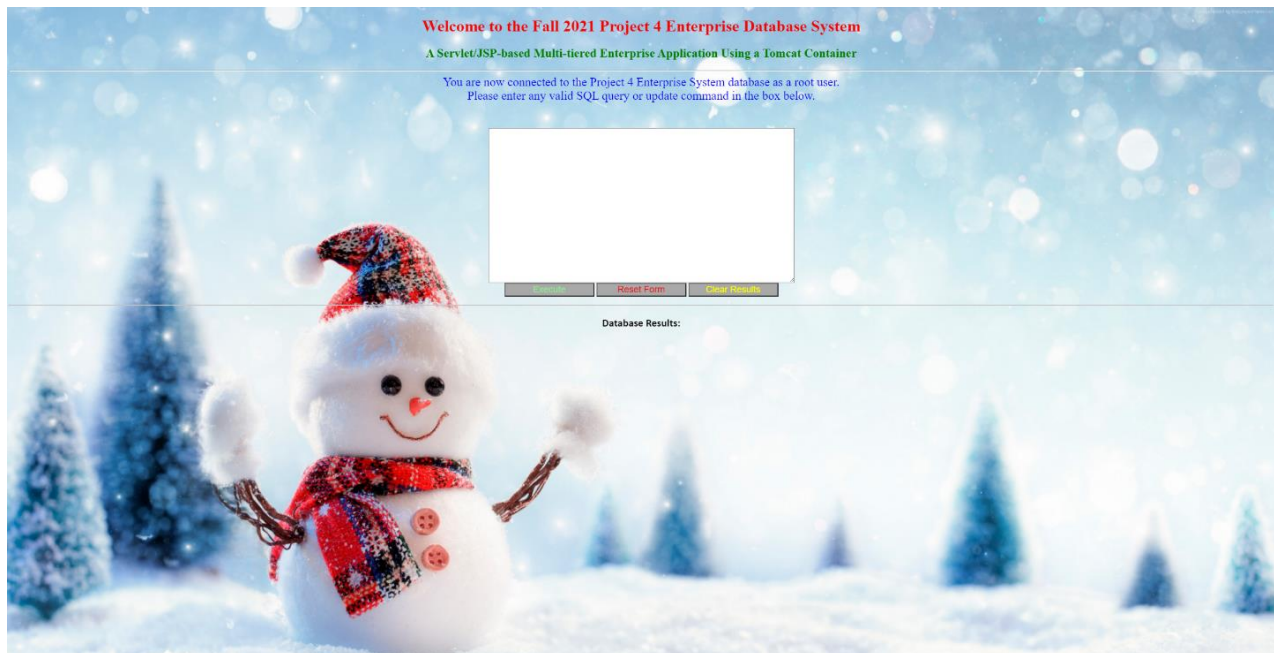
Professor Llewellyn

CNT 4714 Fall 2021

Due 2 Dec 2021

Non-Bonus Version

**#Initial Page:**



# **Command 1:** Query: list the supplier number and supplier name for those suppliers who ship every part.

```
select snum, sname
from suppliers
where not exists
  ( select *
    from parts
    where not exists
      ( select *
        from shipments
        where shipments.snum = suppliers.snum
          and shipments.pnum = parts.pnum
      )
  )
);
```

Execute

Reset Form

Clear Results

Database Results:

snum	sname
S6	Nicola Gianniberti

# **Command 2** is a multi-part sequence that will trigger the business logic.

# **Command 2A:** Query: list all supplier information.

```
select *  
from suppliers;
```

Execute

Reset Form

Clear Results

**Database Results:**

snum	sname	status	city
S1	Michael Schumacher	1	Berlin
S10	David Coulthard	2	London
S11	Bernard Hinault	7	Paris
S12	Eddy Merckx	1	Brussels
S13	Candice Swanepoel	3	Cape Town
S14	Adriana Lima	4	Sao Paulo
S15	Jennifer Lawrence	6	Owensboro
S16	Fernando Alonso	4	Madrid
S17	Rubens Barichello	3	Sao Paulo
S18	Tom Boonen	2	Brussels
S19	Johan Messeuw	1	Eekloo
S2	Juan Pablo Montoya	4	Interlagos
S20	Danilo Rossi	2	Milan
S21	Lizzie Armistead	1	Hempstead
S22	Jan Ullrich	5	Bonn
S3	Dietrich Thurau	1	Berlin
S32	Bernd Schnieder	2	Berlin
S33	Rolf Aldag	3	Berlin
S4	Mark Webber	5	Melbourne
S44	Beryl Burton	4	London
S5	Jenson Button	4	London
S56	Marianne Vos	8	Zandvoort
S6	Nicola Gianniberti	2	Milan
S7	Christian Albers	3	Orlando
S8	Giancarlo Fisichella	3	Milan
S9	Kimi Rikonnen	2	Helsinki

The second part performs the update and causes the business logic to trigger.

# **Command 2B:** add a new record to shipments table (S5, P6, J7, 400)

```
insert into shipments  
values ('S5', 'P6', 'J7', 400);
```

Execute

Reset Form

Clear Results

### Database Results:

The SQL statement completed successfully:  
Business Logic Detected! - Updating Suppliers Status

# **Command 2C:** list all supplier information.

```
select *  
from suppliers;
```

Execute

Reset Form

Clear Results

**Database Results:**

snum	sname	status	city
S1	Michael Schumacher	6	Berlin
S10	David Coulthard	2	London
S11	Bernard Hinault	7	Paris
S12	Eddy Merckx	6	Brussels
S13	Candice Swanepoel	3	Cape Town
S14	Adriana Lima	4	Sao Paulo
S15	Jennifer Lawrence	6	Owensboro
S16	Fernando Alonso	4	Madrid
S17	Rubens Barichello	8	Sao Paulo
S18	Tom Boonen	2	Brussels
S19	Johan Messeuw	1	Eekloo
S2	Juan Pablo Montoya	4	Interlagos
S20	Danilo Rossi	2	Milan
S21	Lizzie Armistead	6	Hempstead
S22	Jan Ullrich	10	Bonn
S3	Dietrich Thurau	6	Berlin
S32	Bernd Schnieder	2	Berlin
S33	Rolf Aldag	3	Berlin
S4	Mark Webber	5	Melbourne
S44	Beryl Burton	9	London
S5	Jenson Button	9	London
S56	Marianne Vos	8	Zandvoort
S6	Nicola Gianniberti	7	Milan
S7	Christian Albers	3	Orlando
S8	Giancarlo Fisichella	3	Milan
S9	Kimi Rikonnen	2	Helsinki

# **Command 3** is a multi-part that does not cause the business logic to trigger

# **Command 3A** list all supplier information

```
select *  
from suppliers;
```

Execute

Reset Form

Clear Results

**Database Results:**

snum	sname	status	city
S1	Michael Schumacher	6	Berlin
S10	David Coulthard	2	London
S11	Bernard Hinault	7	Paris
S12	Eddy Merckx	6	Brussels
S13	Candice Swanepoel	3	Cape Town
S14	Adriana Lima	4	Sao Paulo
S15	Jennifer Lawrence	6	Owensboro
S16	Fernando Alonso	4	Madrid
S17	Rubens Barichello	8	Sao Paulo
S18	Tom Boonen	2	Brussels
S19	Johan Messeuw	1	Eekloo
S2	Juan Pablo Montoya	4	Interlagos
S20	Danilo Rossi	2	Milan
S21	Lizzie Armistead	6	Hempstead
S22	Jan Ullrich	10	Bonn
S3	Dietrich Thureau	6	Berlin
S32	Bernd Schnieder	2	Berlin
S33	Rolf Aldag	3	Berlin
S4	Mark Webber	5	Melbourne
S44	Beryl Burton	9	London
S5	Jenson Button	9	London
S56	Marianne Vos	8	Zandvoort
S6	Nicola Gianniberti	7	Milan
S7	Christian Albers	3	Orlando
S8	Giancarlo Fisichella	3	Milan
S9	Kimi Rikonnen	2	Helsinki

# **Command 3B:** add a new record to the supplier table (S39, Candice Swanepoel, 10, Cardiff)

```
insert into suppliers  
values ('S39','Candice Swanepoel',10,'Cardiff');
```

Execute

Reset Form

Clear Results

### Database Results:

The SQL statement completed successfully:  
Business Logic Not Triggered!

# Command 3C: list all supplier information.

```
select *  
from suppliers;
```

Execute

Reset Form

Clear Results

Database Results:

snum	sname	status	city
S1	Michael Schumacher	6	Berlin
S10	David Coulthard	2	London
S11	Bernard Hinault	7	Paris
S12	Eddy Merckx	6	Brussels
S13	Candice Swanepoel	3	Cape Town
S14	Adriana Lima	4	Sao Paulo
S15	Jennifer Lawrence	6	Owensboro
S16	Fernando Alonso	4	Madrid
S17	Rubens Barichello	8	Sao Paulo
S18	Tom Boonen	2	Brussels
S19	Johan Messeuw	1	Eekloo
S2	Juan Pablo Montoya	4	Interlagos
S20	Danilo Rossi	2	Milan
S21	Lizzie Armistead	6	Hempstead
S22	Jan Ullrich	10	Bonn
S3	Dietrich Thurau	6	Berlin
S32	Bernd Schnieder	2	Berlin
S33	Rolf Aldag	3	Berlin
S39	Candice Swanepoel	10	Cardiff
S4	Mark Webber	5	Melbourne
S44	Beryl Burton	9	London
S5	Jenson Button	9	London
S56	Marianne Vos	8	Zandvoort
S6	Nicola Gianniberti	7	Milan
S7	Christian Albers	3	Orlando
S8	Giancarlo Fisichella	3	Milan
S9	Kimi Rikonnen	2	Helsinki



# **Command 3D:** add a new record to shipments table (S39, P3, J1, 20)

```
insert into shipments  
values ('S39','P3','J1', 20);
```

Execute

Reset Form

Clear Results

### Database Results:

The SQL statement completed successfully:  
Business Logic Not Triggered!

### # Command 3E: list all supplier information

```
select *  
from suppliers;
```

Execute

Reset Form

Clear Results

#### Database Results:

snum	sname	status	city
S1	Michael Schumacher	6	Berlin
S10	David Coulthard	2	London
S11	Bernard Hinault	7	Paris
S12	Eddy Merckx	6	Brussels
S13	Candice Swanepoel	3	Cape Town
S14	Adriana Lima	4	Sao Paulo
S15	Jennifer Lawrence	6	Owensboro
S16	Fernando Alonso	4	Madrid
S17	Rubens Barichello	8	Sao Paulo
S18	Tom Boonen	2	Brussels
S19	Johan Messeuw	1	Eekloo
S2	Juan Pablo Montoya	4	Interlagos
S20	Danilo Rossi	2	Milan
S21	Lizzie Armistead	6	Hempstead
S22	Jan Ullrich	10	Bonn
S3	Dietrich Thurau	6	Berlin
S32	Bernd Schnieder	2	Berlin
S33	Rolf Aldag	3	Berlin
S39	Candice Swanepoel	10	Cardiff
S4	Mark Webber	5	Melbourne
S44	Beryl Burton	9	London
S5	Jenson Button	9	London
S56	Marianne Vos	8	Zandvoort
S6	Nicola Gianniberti	7	Milan
S7	Christian Albers	3	Orlando
S8	Giancarlo Fisichella	3	Milan
S9	Kimi Rikonnen	2	Helsinki

# **Command 4:** List the snum, sname, and pnum for those suppliers who ship the  
# same part to every job. This is a fairly complex SQL nested query.

```
select distinct suppliers.snum, suppliers.sname, shipments.pnum
from suppliers natural join shipments
where shipments.pnum in
  (select pnum
   from parts
   where not exists
     (select *
      from jobs
      where not exists
        (select *
         from shipments
         where shipments.snum = suppliers.snum
              and shipments.pnum = parts.pnum
              and shipments.jnum = jobs.jnum
        )
     )
  );
```

Execute

Reset Form

Clear Results

### Database Results:

snum	sname	pnum
S1	Michael Schumacher	P8

# **Command 5** is a multipart transaction that will cause the business logic to trigger

# **Command 5A:** List all supplier information

```
select *  
from suppliers;
```

Execute

Reset Form

Clear Results

**Database Results:**

snum	sname	status	city
S1	Michael Schumacher	6	Berlin
S10	David Coulthard	2	London
S11	Bernard Hinault	7	Paris
S12	Eddy Merckx	6	Brussels
S13	Candice Swanepoel	3	Cape Town
S14	Adriana Lima	4	Sao Paulo
S15	Jennifer Lawrence	6	Owensboro
S16	Fernando Alonso	4	Madrid
S17	Rubens Barichello	8	Sao Paulo
S18	Tom Boonen	2	Brussels
S19	Johan Messeuw	1	Eekloo
S2	Juan Pablo Montoya	4	Interlagos
S20	Danilo Rossi	2	Milan
S21	Lizzie Armistead	6	Hempstead
S22	Jan Ullrich	10	Bonn
S3	Dietrich Thurau	6	Berlin
S32	Bernd Schnieder	2	Berlin
S33	Rolf Aldag	3	Berlin
S39	Candice Swanepoel	10	Cardiff
S4	Mark Webber	5	Melbourne
S44	Beryl Burton	9	London
S5	Jenson Button	9	London
S56	Marianne Vos	8	Zandvoort
S6	Nicola Gianniberti	7	Milan
S7	Christian Albers	3	Orlando
S8	Giancarlo Fisichella	3	Milan
S9	Kimi Rikonnen	2	Helsinki

# # Command 5B: List all shipment information

```
select *  
from shipments;
```

Execute

Reset Form

Clear Results

## Database Results:

snum	phum	jnum	quantity
S1	P1	J1	40
S1	P1	J4	200
S1	P19	J13	200
S1	P22	J5	40
S1	P3	J13	95
S1	P8	J1	80
S1	P8	J13	200
S1	P8	J2	100
S1	P8	J22	340
S1	P8	J3	200
S1	P8	J4	100
S1	P8	J5	225
S1	P8	J6	230
S12	P3	J22	750
S17	P3	J2	800
S2	P3	J1	40
S2	P3	J2	15
S2	P3	J22	50
S2	P3	J3	20
S2	P3	J4	50
S2	P3	J5	91
S2	P4	J1	50
S21	P2	J4	200
S22	P1	J3	120
S3	P10	J4	200
S3	P24	J2	200
S3	P3	J1	99
S3	P3	J6	75
S3	P4	J6	80
S3	P5	J6	120
S3	P9	J1	110
S39	P3	J1	20
S4	P6	J2	50
S44	P44	J2	100
S44	P48	J5	40
S5	P6	J7	400
S6	P1	J2	40
S6	P10	J3	100
S6	P11	J2	40
S6	P12	J4	75
S6	P13	J5	100
S6	P19	J5	50
S6	P2	J2	60
S6	P24	J2	20
S6	P3	J4	500
S6	P33	J4	40
S6	P4	J2	120
S6	P44	J4	20
S6	P48	J2	100
S6	P5	J22	200
S6	P53	J1	25
S6	P6	J22	120
S6	P7	J1	300
S6	P8	J3	110
S6	P9	J4	180
S7	P1	J2	40
S7	P12	J2	40
S7	P6	J2	40

# **Command 5C:** Update the shipments table by increasing the quantity by 10 every  
# shipment of part P3.

```
update shipments  
set quantity = quantity + 10  
where pnum = 'P3';
```

Execute

Reset Form

Clear Results

### Database Results:

The SQL statement completed successfully:  
Business Logic Detected! - Updating Suppliers Status

## # Command 5D: List all shipment information

```
select *  
from shipments;
```

Execute

Reset Form

Clear Results

Database Results:

snum	pnum	jnum	quantity
S1	P1	J1	40
S1	P1	J4	200
S1	P19	J13	200
S1	P22	J5	40
S1	P3	J13	105
S1	P8	J1	80
S1	P8	J13	200
S1	P8	J2	100
S1	P8	J22	340
S1	P8	J3	200
S1	P8	J4	100
S1	P8	J5	225
S1	P8	J6	230
S12	P3	J22	760
S17	P3	J2	810
S2	P3	J1	50
S2	P3	J2	25
S2	P3	J22	60
S2	P3	J3	30
S2	P3	J4	60
S2	P3	J5	101
S2	P4	J1	50
S21	P2	J4	200
S22	P1	J3	120
S3	P10	J4	200
S3	P24	J2	200
S3	P3	J1	109
S3	P3	J6	85
S3	P4	J6	80
S3	P5	J6	120
S3	P9	J1	110
S39	P3	J1	30
S4	P6	J2	50
S44	P44	J2	100
S44	P48	J5	40
S5	P6	J7	400
S6	P1	J2	40
S6	P10	J3	100
S6	P11	J2	40
S6	P12	J4	75
S6	P13	J5	100
S6	P19	J5	50
S6	P2	J2	60
S6	P24	J2	20
S6	P3	J4	510
S6	P33	J4	40
S6	P4	J2	120
S6	P44	J4	20
S6	P48	J2	100
S6	P5	J22	200
S6	P53	J1	25
S6	P6	J22	120
S6	P7	J1	300
S6	P8	J3	110
S6	P9	J4	180
S7	P1	J2	40
S7	P12	J2	40
S7	P6	J2	40

### # Command 5E: List all supplier information

```
select *  
from suppliers;
```

Execute

Reset Form

Clear Results

#### Database Results:

snum	sname	status	city
S1	Michael Schumacher	11	Berlin
S10	David Coulthard	2	London
S11	Bernard Hinault	7	Paris
S12	Eddy Merckx	11	Brussels
S13	Candice Swanepoel	3	Cape Town
S14	Adriana Lima	4	Sao Paulo
S15	Jennifer Lawrence	6	Owensboro
S16	Fernando Alonso	4	Madrid
S17	Rubens Barichello	13	Sao Paulo
S18	Tom Boonen	2	Brussels
S19	Johan Messeuw	1	Eekloo
S2	Juan Pablo Montoya	9	Interlagos
S20	Danilo Rossi	2	Milan
S21	Lizzie Armistead	11	Hempstead
S22	Jan Ullrich	15	Bonn
S3	Dietrich Thurau	11	Berlin
S32	Bernd Schnieder	2	Berlin
S33	Rolf Aldag	3	Berlin
S39	Candice Swanepoel	10	Cardiff
S4	Mark Webber	5	Melbourne
S44	Beryl Burton	14	London
S5	Jenson Button	14	London
S56	Marianne Vos	8	Zandvoort
S6	Nicola Gianniberti	12	Milan
S7	Christian Albers	3	Orlando
S8	Giancarlo Fisichella	3	Milan
S9	Kimi Rikonnen	2	Helsinki



# **Command 6:** List the snum, and sname for those suppliers who ship only green parts.

# Output should list only supplier number S44

```
select sname
from suppliers
where snum in (select snum
               from shipments
               where pnum in
                   (select pnum
                    from parts
                    where color = 'green'
                   )
               )
and
snum not in (select snum
             from shipments
             where pnum in
                 (select pnum
                  from parts
                  where
color <> 'green'
                 )
            );
```

Execute

Reset Form

Clear Results

### Database Results:

sname
Beryl Burton

# **Command 7:** This will generate an intentional error in command syntax.

# Output should show the error message generated by catching an SQLException

```
select alpha  
from jobs;
```

Execute

Reset Form

Clear Results

### Database Results:

Error executing the SQL statement:  
Unknown column 'alpha' in 'field list'

# **Command 8:** This will not trigger the business logic.

# Output should show the message indicating that the business logic has not triggered

```
update jobs
set city = "Madrid"
where city = "Paris"
```

**Execute** **Reset Form** **Clear Results**

**Database Results:**

The SQL statement completed successfully:  
Business Logic Not Triggered!

# **Command 9:** Verify that command #8 did in fact update the jobs table

```
select *  
from jobs;
```

Execute

Reset Form

Clear Results

### Database Results:

jnum	jname	numworkers	city
J1	Operation DB	45	Berlin
J13	Night Strike	350	Madrid
J2	Really Big Job	500	Melbourne
J22	Project On-Time	200	London
J3	Small Job	100	Chicago
J4	New Job	50	Berlin
J5	My Job	1	Orlando
J6	A New Job	14	Milan

# **Command 10:** This is a basic select command using an aggregation operator

# Business logic not triggered and no table is updated.

```
select count(*) as total_shipments  
from shipments;
```

Execute

Reset Form

Clear Results

Database Results:

total\_shipments

58

## #Web-Apps Project 4 Folder View:

