

Server hostname: cmpstudb-01.cmp.uea.ac.uk Database Name: UEA Username Username: UEA
Username Password: UEA Password

Database Assignment Two – Part 1 – Database definition and loading

Events Table:

```
CREATE TABLE event (  
    ecode CHAR(4) NOT NULL PRIMARY KEY,  
    edesc VARCHAR(20) NOT NULL,  
    elocation VARCHAR(20) NOT NULL,  
    edate DATE NOT NULL,  
    CHECK (edate >= '2019-04-01' AND edate <= '2019-04-30'),  
    etime TIME NOT NULL,  
    CHECK (etime >= '09:00:00'),  
    emax SMALLINT NOT NULL,  
    CHECK (emax >= 1 AND emax <= 1000)  
);
```

Event table constraints

```
xcy20bnu=> \d event  
Table "public.event"  
  Column      |      Type      | Collation | Nullable | Default  
-----+-----+-----+-----+-----  
ecode         | character(4)    |           | not null |  
edesc         | character varying(20) |         | not null |  
elocation     | character varying(20) |         | not null |  
edate         | date            |         | not null |  
etime        | time without time zone |         | not null |  
emax         | smallint        |         | not null |  
Indexes:  
    "event_pkey" PRIMARY KEY, btree (ecode)  
Check constraints:  
    "event_edate_check" CHECK (edate >= '2019-04-01'::date AND edate <= '2019-04-30'::date)  
    "event_emax_check" CHECK (emax >= 1 AND emax <= 1000)  
    "event_etime_check" CHECK (etime >= '09:00:00'::time without time zone)  
xcy20bnu=>
```

The entity integrity constraints in this table are that a primary key exists with no null values.

Constraints:

Domain Name	Constraints
Ecode	Type: Character Max character: 4 Nullable: Not null
Edesc	Type: Character varying Max character: 20 Nullable: Not null
Elocation	Type: Character varying Max characters: 20 Nullable: Not null
Edate	Type: Date Nullable: Not null Check: Date is within April 2019
Etime	Type: Time Nullable: Not null Check: Time is 09:00 or after
Emax	Type: Smallint Nullable: Not null Check: Is between 1 and 1000

Constraint testing:

Wrong date test:

```
xcy20bnu=> INSERT INTO event(ecode, edesc, elocation, edate, etime, emax) VALUES('U435', 'Unicycle', 'Berlin', '2019-03-29', '24:00', 150);
ERROR:  new row for relation "event" violates check constraint "event_edate_check"
DETAIL:  Failing row contains (U435, Unicycle, Berlin, 2019-03-29, 24:00:00, 150).
xcy20bnu=>
```

Wrong time test:

```
xcy20bnu=> INSERT INTO event(ecode, edesc, elocation, edate, etime, emax) VALUES('U435', 'Unicycle', 'Berlin', '2019-04-29', '08:00', 150);
ERROR:  new row for relation "event" violates check constraint "event_etime_check"
DETAIL:  Failing row contains (U435, Unicycle, Berlin, 2019-04-29, 08:00:00, 150).
```

Wrong maximum number of tickets:

```
xcy20bnu=> INSERT INTO event(ecode, edesc, elocation, edate, etime, emax) VALUES('U435', 'Unicycle', 'Berlin', '2019-04-29', '09:00', 0);
ERROR:  new row for relation "event" violates check constraint "event_emax_check"
DETAIL:  Failing row contains (U435, Unicycle, Berlin, 2019-04-29, 09:00:00, 0).
xcy20bnu=> |
```

Event table test data

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('S400', '400m swimming event', 'London', '2019-04-28', '10:00:00', 450);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('S600', '400m swim', 'Stadium 1', '2019-04-01', '09:00:00', 100);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('S900', '400m swim', 'Stadium 1', '2019-04-01', '09:00:00', 100);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('S200', '400m swim', 'Stadium 1', '2019-04-01', '09:00:00', 100);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('I54D', 'lorem id', 'Lodan Wetan', '2019-04-10', '17:55:00', 91);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('U55A', 'imperdiet sapien', 'Perehonivka', '2019-04-12', '14:26:00', 66);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('P55L', 'dolor vel est donec', 'Trzcinica', '2019-04-06', '11:03:00', 62);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('K55Z', 'eu felis', 'Embi', '2019-04-12', '12:01:00', 56);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('I55D', 'consequat metus', 'Bantarsari Kulon', '2019-04-06', '11:03:00', 48);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('N44L', 'et magnis', 'Beverwijk', '2019-04-23', '12:13:00', 95);
```

```
INSERT INTO public.event (ecode, edesc, elocation, edate, etime, emax) VALUES ('P54T', 'nonummy', 'Lapa do Lobo', '2019-04-26', '13:20:00', 28);
```

Spectator table

```
CREATE TABLE spectator (
```

```
sno INTEGER NOT NULL PRIMARY KEY,
```

```
sname VARCHAR(20) NOT NULL,
```

```
semail VARCHAR(100) NOT NULL );
```

Spectator constraints

```
xcy20bnu=> \d spectator
          Table "public.spectator"
  Column |          Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
 sno     | integer                |           | not null |
 sname   | character varying(20)  |           | not null |
 semail  | character varying(20)  |           | not null |
Indexes:
    "spectator_pkey" PRIMARY KEY, btree (sno)

xcy20bnu=> |
```

Sno	Type: Integer Nullable: Not null
Sname	Type: Character varying Max characters: 20 Nullable: Not null
Semail	Type: Character varying Max characters: 20 Nullable: Not null

Testing of constraints:

```
xcy20bnu=> INSERT INTO spectator(sno, sname, semail) VALUES (100, 'Josh Adams', 'josh.adams@gmail.com');
ERROR:  duplicate key value violates unique constraint "spectator_pkey"
DETAIL:  Key (sno)=(100) already exists.
xcy20bnu=> |
```

Test data:

```
INSERT INTO public.spectator (sno, sname, semail) VALUES (2, 'Sam Bagnall', 'sam.bagnall@gmail.com');
```

```
INSERT INTO public.spectator (sno, sname, semail) VALUES (3, 'Anna Bagnall',
'sam.bagnall@gmail.com');
```

```
INSERT INTO public.spectator (sno, sname, semail) VALUES (4, 'Sarah Hodgson',
'sarah.hodgson@gmail.com');
```

```
INSERT INTO public.spectator (sno, sname, semail) VALUES (5, 'Dave', 'd@d.com');
```

```
INSERT INTO public.spectator (sno, sname, semail) VALUES (6, 'David', 'dav@fa.com');
```

```
INSERT INTO public.spectator (sno, sname, semail) VALUES (7, 'Dave', 'd@d.com');
```

```
INSERT INTO public.spectator (sno, sname, semail) VALUES (8, 'David', 'dav@fa.com');
```

Ticket table

```
CREATE TABLE ticket (
tno INTEGER NOT NULL PRIMARY KEY,
ecode CHAR(4) NOT NULL REFERENCES event (ecode) ,
sno INTEGER NOT NULL REFERENCES spectator (sno)
);
```

Ticket table constraints:

```
Table "public.ticket"
Column |      Type      | Collation | Nullable | Default
-----|-----|-----|-----|-----
tno     | integer        |           | not null |
ecode   | character(4)   |           | not null |
sno     | integer        |           | not null |
Indexes:
    "ticket_pkey" PRIMARY KEY, btree (tno)
Check constraints:
    "p_key_over_1" CHECK (tno > 0)
xcy20bnu=> |
```

Tno	Type: Integer
-----	---------------

	Nullable: Not null Must be 1 or over
Ecode	Type: Character Max characters: 4 Nullable: Not null
Sno	Type: integer Nullable: Not null

Testing of constraints:

```
xcy20bnu=> INSERT INTO ticket (tno, ecode, sno) VALUES (0, 'B300', 23);
ERROR:  new row for relation "ticket" violates check constraint "p_key_over_1"
DETAIL:  Failing row contains (0, B300, 23).
xcy20bnu=>
```

```
xcy20bnu=> INSERT INTO ticket (tno, ecode, sno) VALUES (1, 'B300', 23);
ERROR:  duplicate key value violates unique constraint "ticket_pkey"
DETAIL:  Key (tno)=(1) already exists.
xcy20bnu=>
```

Test data:

```
INSERT INTO public.ticket (tno, ecode, sno) VALUES (2, 'S200', 2);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (3, 'S400', 4);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (4, 'P55L', 4);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (5, 'S600', 3);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (6, 'S600', 3);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (7, 'P55L', 5);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (8, 'S200', 8);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (9, 'S400', 7);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (10, 'P55L', 7);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (11, 'S600', 8);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (12, 'S600', 6);
INSERT INTO public.ticket (tno, ecode, sno) VALUES (13, 'P55L', 6);
```

Cancel table

```
CREATE TABLE cancel (
tno INTEGER NOT NULL,
ecode CHAR(4) NOT NULL REFERENCES event (ecode),
sno INTEGER NOT NULL REFERENCES spectator (sno),
```

```

cdate TIMESTAMP NOT NULL DEFAULT NOW(),
cuser VARCHAR(128) NOT NULL DEFAULT 1
);

INSERT INTO public.cancel (tno, ecode, sno, cdate, cuser) VALUES (50, 'P54T', 8, '2019-04-06', 1);
INSERT INTO public.cancel (tno, ecode, sno, cdate, cuser) VALUES (51, 'P54T', 6, '2019-04-07', 1);
INSERT INTO public.cancel (tno, ecode, sno, cdate, cuser) VALUES (52, 'S400', 4, '2019-04-08', 1);
INSERT INTO public.cancel (tno, ecode, sno, cdate, cuser) VALUES (53, 'P55L', 3, '2019-04-09', 1);
INSERT INTO public.cancel (tno, ecode, sno, cdate, cuser) VALUES (54, 'P55L', 8, '2019-04-09', 1);

```

```

CREATE OR REPLACE FUNCTION delete_ticket_from_ticket() RETURNS trigger AS
$BODY$
BEGIN
INSERT INTO cancel (tno, ecode, sno, cdate, cuser)
VALUES (OLD.tno, old.ecode, old.sno, now(), 1);
RETURN OLD;
END;
$BODY$
LANGUAGE plpgsql;

```

```

CREATE TRIGGER delete_from_ticket
BEFORE DELETE
ON ticket
FOR EACH ROW
EXECUTE PROCEDURE delete_ticket_from_ticket();

```

■ OPTIONAL FUNCTION BASED ON TEST DATA STRUCTURE

```

CREATE OR REPLACE FUNCTION add_ticket_to_cancel() RETURNS trigger AS
$$
BEGIN

```

```

IF EXISTS (
    SELECT ticket.tno
    FROM ticket
    WHERE ticket.tno = NEW.tno
) THEN
DELETE FROM ticket WHERE NEW.tno = ticket.tno;
RETURN NEW;
END IF;
RETURN NEW;

END
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER add_ticket_to_cancel
AFTER INSERT
ON cancel
FOR EACH ROW
EXECUTE PROCEDURE add_ticket_to_cancel();

```

Function makes it so that every time a ticket is deleted, it is automatically added to the cancel table.

```

xycy20bnu=> \d cancel
          Table "public.cancel"
  Column |          Type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
 tno     | integer                |           | not null |
 ecode   | character(4)           |           | not null |
 sno     | integer                |           | not null |
 cdate   | timestamp without time zone |         | not null |
 cuser   | character varying(128) |           | not null |
Indexes:
    "cancel_pkey" PRIMARY KEY, btree (tno)

```

Task 2: Tests (Original table values can be seen in the .sql file to see comparisons)

TASK A)

TEST 1 PASS: INSERT INTO spectator (sno, sname, semail) VALUES (100, 'A. N.', 'ano@somewhere.net');

```
pirean=# INSERT INTO spectator (sno, sname, semail) VALUES (100, 'A. N.', 'ano@somewhere.net');
INSERT 0 1
pirean=#
```

TEST 2 PASS INSERT INTO spectator (sno, sname, semail) VALUES (19, 'Anna Bagnall', 'angrybee@gmail.com');

```
pirean=# INSERT INTO spectator (sno, sname, semail) VALUES (19, 'Anna Bagnall', 'angrybee@gmail.com');
INSERT 0 1
pirean=# select * from spectator;
```

TEST 3 FAIL (tno already exists) INSERT INTO spectator (sno, sname, semail) VALUES (100, 'Amy Barret', 'a.b@bb.com');

```
pirean=# INSERT INTO spectator (sno, sname, semail) VALUES (100, 'Amy Barret', 'a.b@bb.com');
ERROR: duplicate key value violates unique constraint "spectator_pkey"
DETAIL: Key (sno)=(100) already exists.
pirean=#
```

Final spectator table for comparison

```
pirean=# select * from spectator;
 sno |      sname      |      semail
-----+-----+-----
  2 | Sam Bagnall     | sam.bagnall@gmail.com
  3 | Anna Bagnall    | anna.bagnall@gmail.com
  4 | Sarah Hodgson   | sarah.hodgson@gmail.com
  5 | Dave            | d@d.com
  6 | David           | dav@fa.com
  7 | Dave            | d@d.com
  8 | David           | dav@fa.com
 100 | A. N.           | ano@somewhere.net
  19 | Anna Bagnall    | angrybee@gmail.com
(9 rows)

pirean=#
```


TASK B)

TEST 1 PASS: INSERT INTO event (ecode, edesc, elocation, edate, etime, emax) VALUES ('A100', '100 metres sprint', 'Stadium 1', '2019-04-19', '14:00', 1000);

```
pirean=# INSERT INTO event (ecode, edesc, elocation, edate, etime, emax) VALUES ('A100', '100 metres sprint', 'Stadium 1', '2019-04-19', '14:00', 1000);
INSERT 0 1
pirean=#
```

TEST 2 PASS: INSERT INTO event (ecode, edesc, elocation, edate, etime, emax) VALUES ('A300', '300 metres sprint', 'Stadium 3', '2019-04-01', '14:00', 100);

```
pirean=# INSERT INTO event (ecode, edesc, elocation, edate, etime, emax) VALUES ('A300', '300 metres sprint', 'Stadium 3', '2019-04-01', '14:00', 100);
INSERT 0 1
pirean=#
```

TEST 3 FAIL (Duplicate event) INSERT INTO event (ecode, edesc, elocation, edate, etime, emax) VALUES ('A300', '300 metres sprint', 'Stadium 3', '2019-04-01', '14:00', 100);

```
pirean=# INSERT INTO event (ecode, edesc, elocation, edate, etime, emax) VALUES ('A300', '300 metres sprint', 'Stadium 3', '2019-04-01', '14:00', 100);
ERROR: duplicate key value violates unique constraint "event_pkey"
DETAIL: Key (ecode)=(A300) already exists.
pirean=#
```

Final event table for comparison:

```
pirean=# select * from event;
 ecode |          edesc          | elocation |   edate   |   etime   |  emax
-----+-----+-----+-----+-----+-----
S400   | 400m swimming event    | London    | 2019-04-28 | 10:00:00 | 450
S600   | 400m swim               | Stadium 1 | 2019-04-01 | 09:00:00 | 100
S900   | 400m swim               | Stadium 1 | 2019-04-01 | 09:00:00 | 100
S200   | 400m swim               | Stadium 1 | 2019-04-01 | 09:00:00 | 100
I54D   | lorem id                | Lodan Wetan | 2019-04-10 | 17:55:00 | 91
U55A   | imperdiet sapien       | Perehonivka | 2019-04-12 | 14:26:00 | 66
P55L   | dolor vel est donec    | Trzcinica | 2019-04-06 | 11:03:00 | 62
I55D   | consequat metus        | Bantarsari Kulon | 2019-04-06 | 11:03:00 | 48
N44L   | et magnis              | Beverwijk | 2019-04-23 | 12:13:00 | 95
P54T   | nonummy                | Lapa do Lobo | 2019-04-26 | 13:20:00 | 28
A100   | 100 metres sprint      | Stadium 1 | 2019-04-01 | 14:00:00 | 1000
A300   | 300 metres sprint      | Stadium 3 | 2019-04-01 | 14:00:00 | 100
(12 rows)

pirean=#
```

TASK C)

TEST 1 PASS: DELETE FROM spectator s WHERE s.sno = 100 AND s.sno NOT IN (SELECT ticket.sno FROM ticket);

```
pirean=# DELETE FROM spectator s WHERE s.sno = 100 AND s.sno NOT IN (SELECT ticket.sno FROM ticket);
DELETE 1
pirean=# |
```

TEST 2 FAIL (Spectator has already purchased tickets so can not be deleted) DELETE FROM spectator s WHERE s.sno = 4 AND s.sno NOT IN (SELECT ticket.sno FROM ticket);

```
pirean=# DELETE FROM spectator s WHERE s.sno = 4 AND s.sno NOT IN (SELECT ticket.sno FROM ticket);
DELETE 0
pirean=# |
```

TEST 2 FAIL (Spectator does not exist) DELETE FROM spectator s WHERE s.sno = 25435654 AND s.sno NOT IN (SELECT ticket.sno FROM ticket);

```
pirean=# DELETE FROM spectator s WHERE s.sno = 25435654 AND s.sno NOT IN (SELECT ticket.sno FROM ticket);
DELETE 0
```

TEST 4 PASS (Spectator does not have any tickets) DELETE FROM spectator s WHERE s.sno = 19 AND s.sno NOT IN (SELECT ticket.sno FROM ticket);

```
pirean=# DELETE FROM spectator s WHERE s.sno = 19 AND s.sno NOT IN (SELECT ticket.sno FROM ticket);
DELETE 1
pirean=#
```

Final spectator table for comparison:

```
pirean=# select * from spectator;
 sno |      sname      |      semail
-----+-----+-----
  2  | Sam Bagnall     | sam.bagnall@gmail.com
  3  | Anna Bagnall    | anna.bagnall@gmail.com
  4  | Sarah Hodgson   | sarah.hodgson@gmail.com
  5  | Dave            | d@d.com
  6  | David           | dav@fa.com
  7  | Dave            | d@d.com
  8  | David           | dav@fa.com
(7 rows)

pirean=#
```

TASK D)

TEST 1 PASS (Event can be deleted as no tickets have been sold for this event): DELETE FROM event e WHERE e.ecode = 'K55Z' AND e.ecode NOT IN (SELECT ticket.ecode FROM ticket);

```
pirean=# DELETE FROM event e WHERE e.ecode = 'K55Z' AND e.ecode NOT IN (SELECT ticket.ecode FROM ticket);
DELETE 1
pirean=#
```

TEST 2 PASS (No ticket have been sold for the event so can be deleted): DELETE FROM event e WHERE e.ecode = 'N44L' AND e.ecode NOT IN (SELECT ticket.ecode FROM ticket);

```
pirean=# DELETE FROM event e WHERE e.ecode = 'N44L' AND e.ecode NOT IN (SELECT ticket.ecode FROM ticket);
DELETE 1
pirean=#
```

TEST 3 FAIL (Tickets have been sold to the event already so can not be deleted): DELETE FROM event e WHERE e.ecode = 'S200' AND e.ecode NOT IN (SELECT ticket.ecode FROM ticket);

```
pirean=# DELETE FROM event e WHERE e.ecode = 'S200' AND e.ecode NOT IN (SELECT ticket.ecode FROM ticket);
DELETE 0
pirean=#
```

Final event table for comparison

```
pirean=# select * from event;
 ecode |      edesc      |      elocation      |      edate      |      etime      |      emax
-----+-----+-----+-----+-----+-----
S400   | 400m swimming event | London              | 2019-04-28      | 10:00:00        | 450
S600   | 400m swim         | Stadium 1           | 2019-04-01      | 09:00:00        | 100
S900   | 400m swim         | Stadium 1           | 2019-04-01      | 09:00:00        | 100
S200   | 400m swim         | Stadium 1           | 2019-04-01      | 09:00:00        | 100
I54D   | lorem id          | Lodan Wetan         | 2019-04-10      | 17:55:00        | 91
U55A   | imperdiet sapien  | Perehonivka         | 2019-04-12      | 14:26:00        | 66
P55L   | dolor vel est donec | Trzcinica           | 2019-04-06      | 11:03:00        | 62
I55D   | consequat metus   | Bantarsari Kulon    | 2019-04-06      | 11:03:00        | 48
P54T   | nonummy           | Lapa do Lobo        | 2019-04-26      | 13:20:00        | 28
A100   | 100 metres sprint | Stadium 1           | 2019-04-19      | 14:00:00        | 1000
A300   | 300 metrs sprint  | Stadium 3           | 2019-04-01      | 14:00:00        | 100
(11 rows)
```

TASK E)

INSERT INTO ticket (tno, ecode, sno)

SELECT @tno, @ecode, @sno

WHERE -- check event exists

@ecode IN (SELECT ecode FROM event)

AND -- check customer doesn't already have a ticket to that event

NOT EXISTS (

SELECT ecode, sno FROM ticket

WHERE ecode = @ecode AND sno = @sno

)

AND -- check that ticket limit emax has not been reached

((SELECT COUNT(ecode) FROM ticket WHERE ecode = @ecode) <

(SELECT emax FROM event WHERE event.ecode = @ecode));

For these tests I will run the same query but change the variables due to the query length.

TEST 1 PASS (Event exists and customer does not already have a ticket). tno = 20, ecode = S400, sno = 3

```
pirean=# INSERT INTO ticket (tno, ecode, sno)
pirean=# SELECT 20, 'S400', 3
pirean=# WHERE
pirean=# 'S400' IN (SELECT ecode FROM event)
pirean=# AND
pirean=# NOT EXISTS (
pirean(# SELECT ecode, sno FROM ticket
pirean(# WHERE ecode = 'S400' AND sno = 3
pirean(# )
pirean=# AND
pirean=# ((SELECT COUNT(ecode) FROM ticket WHERE ecode = 'S400') <
pirean(# (SELECT emax FROM event WHERE event.ecode = 'S400'));
INSERT 0 1
pirean=#
```

TEST 2 FAIL (Event exists but customer already has a ticket to this event) tno = 21, ecode = S400, sno = 7

```
pirean=# INSERT INTO ticket (tno, ecode, sno)
pirean=# SELECT 21, 'S400', 7
pirean=# WHERE
pirean=# 'S400' IN (SELECT ecode FROM event)
pirean=# AND
pirean=# NOT EXISTS (
pirean(# SELECT ecode, sno FROM ticket
pirean(# WHERE ecode = 'S400' AND sno = 7
pirean(# )
pirean=# AND
pirean=# ((SELECT COUNT(ecode) FROM ticket WHERE ecode = 'S400') <
pirean(# (SELECT emax FROM event WHERE event.ecode = 'S400'));
INSERT 0 0
pirean=#
```

TEST 3 FAIL (Event does not exist so a ticket can not be issued) tno = 23, ecode = SZZZ, sno = 6

```
pirean=# INSERT INTO ticket (tno, ecode, sno)
pirean=# SELECT 23, 'SZZZ', 6
pirean=# WHERE
pirean=# 'SZZZ' IN (SELECT ecode FROM event)
pirean=# AND
pirean=# NOT EXISTS (
pirean=#     SELECT ecode, sno FROM ticket
pirean=#     WHERE ecode = 'SZZZ' AND sno = 6
pirean=# )
pirean=# AND
pirean=# ((SELECT COUNT(ecode) FROM ticket WHERE ecode = 'SZZZ') <
pirean=# (SELECT emax FROM event WHERE event.ecode = 'SZZZ'));
INSERT 0 0
pirean=#
```

Final ticket table for comparison

```
pirean=# select * from ticket;
 tno | ecode | sno
-----+-----+-----
  2 | S200 |  2
  3 | S400 |  4
  4 | P55L |  4
  5 | S600 |  3
  6 | S600 |  3
  7 | P55L |  5
  8 | S200 |  8
  9 | S400 |  7
 10 | P55L |  7
 11 | S600 |  8
 12 | S600 |  6
 13 | P55L |  6
 20 | S400 |  3
(13 rows)
```

TASK P)

```
CREATE OR REPLACE VIEW viewp AS SELECT edate, elocation, COUNT(ticket.ecode) FROM event LEFT
JOIN ticket ON event.ecode = ticket.ecode GROUP BY edate, elocation;
SELECT * FROM viewp;
```

```
pirean=# select * from viewp
pirean=# ;
  edate   |   elocation   | count
-----+-----+-----
2019-04-06 | Trzcinica     |      4
2019-04-26 | Lapa do Lobo  |      0
2019-04-28 | London        |      3
2019-04-10 | Lodan Wetan   |      0
2019-04-06 | Bantarsari Kulon |      0
2019-04-12 | Perehonivka   |      0
2019-04-01 | Stadium 1     |      6
2019-04-19 | Stadium 1     |      0
2019-04-01 | Stadium 3     |      0
(9 rows)

pirean=#
```

TASK Q)

```
CREATE OR REPLACE VIEW viewq AS SELECT ticket.ecode, event.edesc, COUNT(tno) FROM ticket LEFT JOIN event ON ticket.ecode = event.ecode GROUP BY ticket.ecode, event.edesc ORDER BY edesc;
SELECT * FROM viewq;
```

```
pirean=# SELECT * FROM viewq;
ecode |          edesc          | count
-----+-----+-----
S200  | 400m swim              |      2
S600  | 400m swim              |      4
S400  | 400m swimming event    |      3
P55L  | dolor vel est donec    |      4
(4 rows)

pirean=#
```

TASK R)

```
TEST 1 PASS SELECT ticket.ecode, event.edesc, COUNT(tno) FROM ticket LEFT JOIN event ON ticket.ecode = event.ecode WHERE ticket.ecode = 'S400' GROUP BY ticket.ecode, event.edesc;
```

```
pirean=# SELECT ticket.ecode, event.edesc, COUNT(tno) FROM ticket LEFT JOIN event ON ticket.ecode = event.ecode WHERE ticket.ecode = 'S400' GROUP BY ticket.ecode, event.edesc;
ecode |          edesc          | count
-----+-----+-----
S400  | 400m swimming event    |      3
(1 row)

pirean=#
```

```
TEST 2 PASS SELECT ticket.ecode, event.edesc, COUNT(tno) FROM ticket LEFT JOIN event ON ticket.ecode = event.ecode WHERE ticket.ecode = 'S600' GROUP BY ticket.ecode, event.edesc;
```

```
pirean=# SELECT ticket.ecode, event.edesc, COUNT(tno) FROM ticket LEFT JOIN event ON ticket.ecode = event.ecode WHERE ticket.ecode = 'S600' GROUP BY ticket.ecode, event.edesc;
ecode |          edesc          | count
-----+-----+-----
S600  | 400m swim              |      4
(1 row)

pirean=#
```

TASK 5)

DROP TABLE sitinerary;

CREATE TABLE IF NOT EXISTS sitinerary (

sno integer NOT NULL REFERENCES spectator(sno),

sname VARCHAR(20) NOT NULL,

ecode VARCHAR(20),

edate DATE NOT NULL,

elocation VARCHAR(20) NOT NULL,

etime TIME NOT NULL,

edesc VARCHAR(20) NOT NULL

);

INSERT INTO sitinerary (sno, sname, ecode, edate, elocation, etime, edesc)

SELECT ticket.sno, spectator.sname, event.ecode, event.edate, event.elocation, event.etime,
event.edesc

FROM ticket

INNER JOIN event ON ticket.ecode = event.ecode

INNER JOIN spectator ON ticket.sno = spectator.sno

WHERE ticket.sno = 4;

```
pirean=# DROP TABLE sitinerary;
DROP TABLE
pirean=# CREATE TABLE IF NOT EXISTS sitinerary (
pirean(#      sno integer NOT NULL REFERENCES spectator(sno),
pirean(#      sname VARCHAR(20) NOT NULL,
pirean(#      ecode VARCHAR(20),
pirean(#      edate DATE NOT NULL,
pirean(#      elocation VARCHAR(20) NOT NULL,
pirean(#      etime TIME NOT NULL,
pirean(#      edesc VARCHAR(20) NOT NULL
pirean(# );
CREATE TABLE
pirean=#
pirean=#
pirean=# INSERT INTO sitinerary (sno, sname, ecode, edate, elocation, etime, edesc)
pirean=#
pirean=# SELECT ticket.sno, spectator.sname, event.ecode, event.edate, event.elocation, event.etime, event.edesc
pirean=# FROM ticket
pirean=# INNER JOIN event ON ticket.ecode = event.ecode
pirean=# INNER JOIN spectator ON ticket.sno = spectator.sno
pirean=# WHERE ticket.sno = 4;
INSERT 0 2
```

Successful insert to new table sitinerary

```

pirean=# select * from sitinerary where sno = 4;
sno |      sname      | ecode |   edate   | elocation |   etime   |      edesc      |
-----+-----+-----+-----+-----+-----+-----
  4 | Sarah Hodgson | S400  | 2019-04-28 | London    | 10:00:00 | 400m swimming event
  4 | Sarah Hodgson | P55L  | 2019-04-06 | Trzcinica | 11:03:00 | dolor vel est donec
(2 rows)

pirean=#

```

TASK T)

-- Creat a function that will check whether the ticket is valid or has been cancelled. This is done by checking whether the ticket exists in the ticket table or not. If it does not exist in the ticket table, it is presumed to be cancelled and therefore invalid.

```
CREATE OR REPLACE FUNCTION check_ticket_in_ticket(tno int)
```

```
RETURNS BOOL AS
```

```
$$
```

```
BEGIN
```

```
    IF EXISTS (SELECT ticket.tno FROM ticket WHERE ticket.tno = $1) THEN
```

```
        RETURN TRUE;
```

```
    ELSE RETURN FALSE;
```

```
    END IF;
```

```
    RETURN TRUE;
```

```
END;
```

```
$$
```

```
LANGUAGE plpgsql;
```

-- Create a view that contains every ticket by using SQL union. This view also uses the above function to state whether the ticket is valid or not.

```
CREATE OR REPLACE VIEW all_tickets AS
```

```
SELECT tno, ecode, sno, check_ticket_in_ticket(tno) FROM ticket
```

```
UNION ALL
```



```
SELECT tno, ecode, sno, check_ticket_in_ticket(tno) FROM cancel;
```

-- Create a final view that joins the above view to the spectator table in order to get the spectator's name.

```
CREATE OR REPLACE VIEW all_tickets_with_info AS
```

```
SELECT all_tickets.tno, all_tickets.ecode, all_tickets.sno, all_tickets.check_ticket_in_ticket,  
spectator.sname
```

```
FROM all_tickets
```

```
LEFT JOIN spectator ON all_tickets.sno = spectator.sno;
```

TEST 1: SELECT sname, ecode, check_ticket_in_ticket as ticket_status FROM all_tickets_with_info
WHERE tno = 5;

```
pirean=# SELECT sname, ecode, check_ticket_in_ticket as ticket_status FROM all_tickets_with_info WHERE tno = 5;  
   sname   | ecode | ticket_status  
-----+-----+-----  
 Anna Bagnall | P55L  | t  
(1 row)  
  
pirean=#
```

TEST 2: SELECT sname, ecode, check_ticket_in_ticket as ticket_status FROM all_tickets_with_info
WHERE tno = 51;

```
pirean=# SELECT sname, ecode, check_ticket_in_ticket as ticket_status FROM all_tickets_with_info WHERE tno = 51;  
   sname   | ecode | ticket_status  
-----+-----+-----  
 David   | P54T  | f  
(1 row)  
  
pirean=#
```

TEST 3: SELECT sname, ecode, check_ticket_in_ticket as ticket_status FROM all_tickets_with_info
WHERE tno = 52;

```
pirean=# SELECT sname, ecode, check_ticket_in_ticket as ticket_status FROM all_tickets_with_info WHERE tno = 52;  
   sname   | ecode | ticket_status  
-----+-----+-----  
 Sarah Hodgson | S400  | f  
(1 row)  
  
pirean=#
```

Final ticket tatus table (multiple customers have multiple tickets so names are similar)

```
pirean=# select * from all_tickets_with_info;
 tno | ecode | sno | check_ticket_in_ticket |      sname
-----+-----+-----+-----+-----
  2 | S200 |  2 | t | Sam Bagnall
  3 | S400 |  4 | t | Sarah Hodgson
  4 | S600 |  4 | t | Sarah Hodgson
  5 | S600 |  3 | t | Anna Bagnall
  6 | P55L |  3 | t | Anna Bagnall
  7 | P55L |  5 | t | Dave
  9 | S400 |  7 | t | Dave
 10 | P55L |  7 | t | Dave
 11 | S600 |  8 | t | David
 12 | S600 |  6 | t | David
 13 | P55L |  6 | t | David
 20 | S400 |  3 | t | Anna Bagnall
 50 | P55L |  8 | f | David
 51 | P54T |  6 | f | David
 52 | S400 |  4 | f | Sarah Hodgson
 53 | P55L |  3 | f | Anna Bagnall
 54 | P55L |  8 | f | David
(17 rows)
```

TASK V)

TEST 1: SELECT tno, ecode, sno, check_ticket_in_ticket as ticket_status, sname FROM all_tickets_with_info WHERE ecode = 'S400' AND check_ticket_in_ticket = false;

```
pirean=# SELECT tno, ecode, sno, check_ticket_in_ticket as ticket_status, sname FROM all_tickets_with_info WHERE ecode = 'S
400' AND check_ticket_in_ticket = false;
 tno | ecode | sno | ticket_status |      sname
-----+-----+-----+-----+-----
  52 | S400 |  4 | f | Sarah Hodgson
(1 row)

pirean=#
```

TEST 2 PASS: SELECT tno, ecode, sno, check_ticket_in_ticket as ticket_status, sname FROM all_tickets_with_info WHERE ecode = 'P55L' AND check_ticket_in_ticket = false;

```
pirean=# SELECT tno, ecode, sno, check_ticket_in_ticket as ticket_status, sname FROM all_tickets_with_info WHERE ecode = 'P
55L' AND check_ticket_in_ticket = false;
 tno | ecode | sno | ticket_status |      sname
-----+-----+-----+-----+-----
  53 | P55L |  3 | f | Anna Bagnall
  54 | P55L |  8 | f | David
(2 rows)

pirean=#
```

x)

Run from Python:

```
cur.close()
conn.close()
```

z)

TRUNCATE event CASCADE;

TRUNCATE ticket CASCADE;

TRUNCATE cancel CASCADE;

TRUNCATE spectator CASCADE;

TRUNCATE sitinerary CASCADE;

Messages

```
NOTICE: truncate cascades to table "cancel"  
NOTICE: truncate cascades to table "ticket"  
NOTICE: truncate cascades to table "sitinerary"  
NOTICE: truncate cascades to table "cancel"  
NOTICE: truncate cascades to table "ticket"  
TRUNCATE TABLE
```

Query returned successfully in 76 msec.

Other/misc queries:

```
CREATE OR REPLACE FUNCTION on_insert_ticket_add_to_tablet() RETURNS trigger AS
$BODY$
BEGIN
    INSERT INTO tablet (tno, sname, ecode, status)
    SELECT NEW.tno, sname, NEW.ecode, true
    FROM ticket
    LEFT JOIN spectator ON NEW.sno = spectator.sno;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER on_insert_ticket_add_to_tablet
BEFORE INSERT
ON ticket
FOR EACH ROW
EXECUTE PROCEDURE on_insert_ticket_add_to_tablet();
```

```
INSERT INTO ticket (tno, ecode, sno)
SELECT 43, 'S200', 4
WHERE
'S200' IN (SELECT ecode FROM event)
AND
NOT EXISTS (
```

```
SELECT ecode, sno FROM ticket
WHERE ecode = 'S200' AND sno = 4
)
AND
((SELECT COUNT(ecode) FROM ticket WHERE ecode = 'S200') <
(SELECT emax FROM event WHERE event.ecode = 'S200'));
```