

```

CREATE SCHEMA Assessment_100536383;

SET search_path to Assessment_100536383;

-- Creating Table: Event
CREATE TABLE event (
    ecode CHAR(4) PRIMARY KEY,
    edesc VARCHAR(20) NOT NULL,
    elocation VARCHAR(20) NOT NULL,
    edate DATE NOT NULL CHECK (edate BETWEEN '2026-07-01' AND '2026-07-31'),
    etime TIME NOT NULL CHECK (etime >= '09:00:00'),
    emax SMALLINT NOT NULL CHECK (1001 > emax AND emax > 0)
);

SELECT * FROM event;

-- Creating Table: Spectator
CREATE TABLE spectator (
    sno INTEGER PRIMARY KEY,
    sname VARCHAR(20) NOT NULL,
    semail VARCHAR(20) NOT NULL UNIQUE
);

SELECT * FROM spectator;

--Creating Table: Ticket
CREATE TABLE ticket (
    tno INTEGER PRIMARY KEY,
    ecode CHAR(4) NOT NULL REFERENCES event(ecode) ON DELETE CASCADE,
    sno INTEGER NOT NULL REFERENCES spectator(sno) ON DELETE CASCADE,
    CONSTRAINT only_one_ticket UNIQUE (ecode, sno)
);

SELECT * FROM ticket;

--Creating Table: Cancel
CREATE TABLE cancel (
    tno INTEGER PRIMARY KEY,
    ecode CHAR(4) NOT NULL,
    sno INTEGER NOT NULL,
    cdate TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    cuser VARCHAR(128) NOT NULL
);

SELECT * FROM cancel;

--Creating Function & Trigger: Event Capacity before issuing ticket
CREATE OR REPLACE FUNCTION check_capacity()
RETURNS TRIGGER AS $$

BEGIN
    IF (SELECT COUNT(*) FROM ticket WHERE ecode = NEW.ecode) >=
        (SELECT emax FROM event WHERE ecode = NEW.ecode) THEN
        RAISE EXCEPTION 'Event % is at full capacity (Max : %)', NEW.ecode, (SELECT emax FROM event WHERE ecode = NEW.ecode);
    ELSE
        RETURN NEW;
    END IF;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER target_capacity
BEFORE INSERT ON ticket
FOR EACH ROW
EXECUTE FUNCTION check_capacity();

--Creating Function & Trigger: Cancelled Tickets Logging in to Cancel Table
CREATE OR REPLACE FUNCTION ticket_cancel_alteration()
RETURNS TRIGGER AS $$

BEGIN
    INSERT INTO cancel (tno, ecode, sno, cdate, cuser)
    VALUES (OLD.tno, OLD.ecode, OLD.sno, CURRENT_TIMESTAMP, CURRENT_USER)
    ON CONFLICT (tno) DO NOTHING;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER ticket_cancel_alter
BEFORE DELETE ON ticket
FOR EACH ROW
EXECUTE FUNCTION ticket_cancel_alteration();

--Creating View: Event Schedule for each spectator
CREATE VIEW spectator_event_schedule AS
    SELECT s.sno, s.sname, e.ecode, e.edesc, e.edate, e.etime, e.elocation
    FROM spectator s
    JOIN ticket t ON s.sno = t.sno

```

```

JOIN event e ON t.ecode = e.ecode;

----- My Train Data -----

-- Insert a data in to event table
INSERT INTO event(ecode, edesc, elocation, edate, etime, emax) VALUES
('E001', 'BADMINTON', 'WEMBLEY', '2026-07-15', '14:00', 10),
('E002', 'SWIMMING', 'LLYODS', '2026-07-15', '16:00', 2),
('E003', 'RACING', 'MARINA', '2026-07-16', '10:00', 1000),
('E004', 'JUDO', 'ARENA 2', '2026-07-17', '18:00', 5);

SELECT * FROM event;

-- Insert a data in to spectator table
INSERT INTO spectator(sno, sname, semail) VALUES
(1001, 'Shevangi Rathod', 'shevangi@gmail.com'),
(1002, 'Dhruv R', 'dhruv@gmail.com'),
(1003, 'Yuvraj', 'yuvraj@gmail.com'),
(1004, 'Jayraj', 'jay@gmail.com');

SELECT * FROM spectator;

-- Insert a data in to ticket table
INSERT INTO ticket(tno, ecode, sno) VALUES
(1, 'E001', 1001),
(2, 'E001', 1002),
(3, 'E002', 1001),
(4, 'E003', 1002),
(5, 'E004', 1004);

SELECT * FROM ticket;

-- Ticket Cancelled
DELETE FROM Ticket WHERE tno = 3;

SELECT * FROM TICKET;
SELECT * FROM CANCEL;

-- Primary key violation
INSERT INTO spectator(sno, sname, semail) VALUES
(1001, 'Bob', 'bob@gmail.com');

-- Unique Constraint Violation
INSERT INTO spectator(sno, sname, semail) VALUES
(1005, 'Bob', 'shevangi@gmail.com');

-- Foreign key Violation
INSERT INTO ticket(tno, ecode, sno) VALUES
(99, 'AXXA', 1001);

-- Check Constraint(Domain)
INSERT INTO event(ecode, edesc, elocation, edate, etime, emax) VALUES
('E006', 'RUNNING', 'WEMBLEY', '2026-07-19', '14:00', 0);

-- Duplicate Ticket
INSERT INTO ticket(tno, ecode, sno) VALUES
(10, 'E001', 1001);

-- Trigger : Capacity Check
INSERT INTO ticket(tno, ecode, sno) VALUES
(11, 'E002', 1003),
(12, 'E002', 1004);

SELECT * FROM ticket;

INSERT INTO ticket(tno, ecode, sno) VALUES
(13, 'E002', 1001);

-- Trigger : Delete Spectator
DELETE FROM spectator WHERE sno = 1001;

SELECT * FROM spectator;
SELECT * FROM Ticket;
SELECT * FROM Cancel;

-- Trigger : Delete Event
DELETE FROM event WHERE ecode = 'E003';
SELECT * FROM event;
SELECT * FROM Ticket;

SELECT * FROM Cancel;

```