

Class: CC05Group No: 7

- 1- Mai Hữu Nghĩa (2052612)
- 2- Trần Trí Đạt (2052443)
- 3- Triệu Phúc Nguyên (2011715)
- 4- Trần Trọng Nhân (2053288)

Please feel free to go to our online folder via the following link or scan the QR code:

https://github.com/TravisMai/CO2013_DBS_HK221

Database System – S221
Assignment 2 Report

----o0o----



Content	Student No. ¹	Note ²
PART 1: CREATE DATABASE (4p)		
I. Create table (2.5p)		
- Table 1-10	2	
- Table 11-18	1	
- Table 18-24	3	
- Table 20-24	4	
II. Insert (1.5p)		
- Table 2,3,5,15,16,17,18	1	
- Table 1,3,4,9,11,14,19,20	2	
- Table 2,6,7,8,13,18,19	3	
- Table 10,12,21,22,23,24	4	
PART 2: STORE PROCEDURE, FUNCTION, TRIGGER		
2.1.1		
- A trainee can participate 3 seasons at most (0.5p)	2	
- Cannot participate a season if had joined in a debut night (0.5p)	3	
2.1.2 Calculate the total number of votes of a group (derived attribute of table 19) (0.5 p)	1	
2.1.3 A trainee has at most:	2	

¹ Student(s) who do this task. Student No (1..4) is the same as group member in the top of page 1.

² For lecturer's use only

- 1 group stage in episode 2, 3, 4 (0.25 p) - 1 group stage and one individual stage in episode 5 (0.25 p)	4	
2.2.1 Print list of trainees come to the next episode (or debut if input episode is debut night) (0.5p)	1 3	
2.2.2 Create a procedure/function to retrieve the result of a trainee in a season (0.5p)	2	
PART 3: BUILDING APPLICATIONS (3p)		
I. Create user (0.5p)	1 2	
II. Implement features		
2.1.1 Log in (0.25p)	1	
2.1.2 Log out (0.25p)	2	
2.2.1 Add information for a new trainee (0.5p)	1 2 3	
2.2.2 Search information of trainees by name (0.5p)	1 2 4	
2.2.3. View information of a trainee (0.5p)	1 2	
2.3.4. Retrieve result of a trainee in a season (0.5p)	1 2	
Bonus: - User interface is attractive and friendly (0.5p) - Building a data model (0.5p)	1 2	

PART 1: CREATE DATABASE IN MARIADB (XAMPP)**I. Create table****1. Company (Cnumber, name, address, phone, Edate)**

```
-- Company --
CREATE Table Company(
    Cnumber char(4) not NULL primary key,
    Name text not NULL,
    CAddress text,
    Phone numeric(10) not NULL UNIQUE,
    Edate date
);

-- Trigger check Company number --
DELIMITER //
CREATE OR REPLACE TRIGGER check_CNumber
AFTER INSERT ON company
FOR EACH ROW
BEGIN
    IF (new.Cnumber NOT RLIKE '[C][0-9][0-9][0-9]') THEN
        SIGNAL SQLSTATE '45000' SET
            MESSAGE_TEXT = 'Not correct Company Number';
    END IF;
END;
//
DELIMITER ;

DELIMITER //
CREATE OR REPLACE TRIGGER check_CNumber_Update
AFTER UPDATE ON company
FOR EACH ROW
BEGIN
    IF (new.Cnumber NOT RLIKE '[C][0-9][0-9][0-9]') THEN
        SIGNAL SQLSTATE '45000' SET
            MESSAGE_TEXT = 'Not correct Company Number';
    END IF;
END;
//
DELIMITER ;
```

2. Person (SSN, Fname, Lname, address, phone)

```
-- Person --
CREATE table Person(
    Ssn numeric(12,0) NOT NULL PRIMARY KEY,
    Fname text,
    Lname text,
    PAddress text,
    Phone numeric(10) not NULL UNIQUE
);
```

3. Trainee (SSN, DoB, photo, company_ID)

```
-- Trainee --
CREATE table Trainee(
    Ssn numeric(12,0) NOT NULL PRIMARY KEY,
    DoB date,
    Photo varchar(2083),
    Company_ID char(4),
    foreign key (Ssn) references Person(Ssn) on delete cascade,
    foreign key (Company_ID) references Company(Cnumber) on delete
set null
);
```

4. MC (SSN)

```
-- MC --
CREATE table MC(
    Ssn numeric(12,0) NOT NULL PRIMARY KEY,
    foreign key (Ssn) REFERENCES person(Ssn) on delete CASCADE
);
```

5. Mentor (SSN)

```
-- Mentor --
CREATE table Mentor(
    Ssn numeric(12,0) NOT NULL PRIMARY KEY,
    foreign key (Ssn) references person(ssn) on delete CASCADE
);
```

6. song_sequecing(num)

```
-- Table for auto increament --
CREATE table song_sequecing(
    Num INT NOT NULL AUTO_INCREMENT PRIMARY KEY
);
```

7. Song (number, released_year, name, singer_SSN_fist_performed)

```
-- Table for song --
```

```
CREATE table Song(  
    Number varchar(5) not NULL PRIMARY KEY,  
    Released_year year,  
    Name text,  
    Singer_SSN_first_performed numeric(12,0),  
    FOREIGN KEY (Singer_SSN_first_performed) references  
Singer(Ssn) on DELETE set NULL  
);
```

```
-- Create a trigger for `Song` auto INCREMENT --
```

```
DELIMITER //
```

```
CREATE TRIGGER Song_num_insert  
BEFORE INSERT ON song  
FOR EACH ROW  
BEGIN  
    INSERT INTO song_sequecing VALUES (NULL);  
    SET new.Number = CONCAT("S", LPAD(LAST_INSERT_ID(), 3, '0'));  
END;//  
DELIMITER ;
```

8. ThemeSong (song ID)

```
-- Themesong --
```

```
CREATE table ThemeSong(  
    Song_ID varchar(5) not NULL PRIMARY KEY,  
    FOREIGN KEY (Song_ID) references Song(number) on DELETE cascade  
);
```

9. SongComposedBy (song ID, composer_SSN)

```
-- SongComposedBy --
```

```
CREATE table SongComposedBy(  
    Song_ID varchar(5) not NULL,  
    Composer_Ssn numeric(12,0) not NULL,  
    primary key (Song_ID, Composer_Ssn),  
    FOREIGN KEY (Song_ID) REFERENCES Song(number),  
    FOREIGN KEY (Composer_Ssn) references SongWriter(Ssn)  
);
```

10. Singer (SSN, guest_ID)

```
-- Singer --
CREATE table Singer(
    Ssn numeric(12,0) NOT NULL PRIMARY KEY,
    Guest_ID INT,
    FOREIGN KEY (Ssn) REFERENCES Mentor(Ssn),
    FOREIGN KEY (Guest_ID) references InvitedGuest(guest_ID)
);
```

11. SingerSignatureSong (SSN, song_name)

```
-- SingerSignatureSong --
CREATE table SingerSignatureSong(
    Ssn numeric(12,0) not NULL PRIMARY KEY,
    Song_name text,
    FOREIGN KEY (Ssn) REFERENCES Singer(Ssn)
);
```

12. Producer (SSN)

```
-- Producer --
CREATE table Producer(
    Ssn numeric(12,0) NOT NULL PRIMARY KEY,
    FOREIGN KEY (Ssn) REFERENCES Mentor(Ssn) on DELETE cascade
);
```

13. ProducerProgram (SSN, Program_name)

```
-- ProducerProgram --
CREATE table ProducerProgram(
    Ssn numeric (12,0) not null,
    Program_name varchar(512),
    primary key (Ssn, Program_name),
    foreign key (Ssn) references Producer(Ssn) on DELETE cascade
);
```

14. SongWriter (SSN)

```
-- SongWriter --
CREATE table SongWriter(
    Ssn numeric (12,0) not null,
    primary key (Ssn),
    foreign key (Ssn) references Mentor(Ssn) on DELETE cascade
);
```

15. Season (year, location, themesong_ID, MC_SSN)

```
-- Season --
CREATE table Season(
    SYear year primary key,
    Location text,
    Themesong_ID varchar(5),
    MC_Ssn NUMERIC(12,0) not null,
    FOREIGN KEY (Themesong_ID) REFERENCES ThemeSong(Song_ID) on
delete set NULL,
    FOREIGN key (MC_Ssn) REFERENCES MC(Ssn) on DELETE cascade
);
```

16. SeasonMentor (year, SSN_mentor)

```
-- SeasonMentor --
CREATE TABLE SeasonMentor(
    Syear year not null,
    Ssn_mentor NUMERIC(12,0) not null,
    PRIMARY KEY (Syear, Ssn_mentor),
    FOREIGN KEY (Syear) REFERENCES Season(Syear) on delete cascade,
    FOREIGN KEY (Ssn_mentor) REFERENCES Mentor(Ssn) on delete
cascade
);
```

17. SeasonTrainee (year, SSN_trainee)

```
-- SeansonTrainee --
CREATE TABLE SeasonTrainee(
    Syear year not NULL,
    Ssn_trainee NUMERIC(12,0) not NULL,
    PRIMARY KEY (Syear, Ssn_trainee),
    FOREIGN KEY (Syear) REFERENCES Season(Syear) on delete cascade,
    FOREIGN KEY (Ssn_trainee) REFERENCES Trainee(Ssn) on delete
cascade
);
```

18. MentorValuateTrainee (year, SSN_trainee, SSN_mentor, score)

```
-- MentorValuateTrainee --
CREATE TABLE MentorValuateTrainee(
    Syear year not NULL,
    Ssn_trainee NUMERIC(12,0) not null,
    Ssn_mentor NUMERIC(12,0) not null,
    Score INT(3),
    CONSTRAINT CheckScore CHECK (Score>=0 and Score <=100),
    PRIMARY KEY (Syear, Ssn_mentor, Ssn_trainee),
    FOREIGN KEY (Syear) REFERENCES Season(Syear) on delete cascade,
    FOREIGN KEY(Ssn_trainee)REFERENCES Trainee(Ssn) on delete
cascade,
    FOREIGN KEY (Ssn_mentor) REFERENCES Mentor(Ssn) on delete
cascade
);
```

19. Episode (year, No, name, datetime, duration)

```
-- Episode --
CREATE TABLE Episode(
    Syear year not NULL,
    Ep_No int(1) not NULL CHECK (Ep_No>=1 and Ep_no <=5),
    Name text,
    Datetime DATETIME,
    Duration int(3),
    Primary KEY (Syear, Ep_No),
    FOREIGN KEY (Syear) REFERENCES Season(Syear) on DELETE cascade
);
```


20. Stage (year, ep No, stage No, is_group, skill, total vote, song_ID)

```
-- Stage --
```

```
CREATE TABLE Stage(
    Syear year not NULL,
    Ep_No int(1) not NULL CHECK (Ep_No>=1 and Ep_no <=5),
    Stage_No INT(3) NOT NULL,
    Is_Group BOOLEAN NOT NULL, -- Yes: group, No: individual --
    Skill int(1) default 4 ,
    Total_Votes INT, -- Derived attribute --
    Song_ID varchar(5) NOT NULL,
    CONSTRAINT CheckSkill CHECK (Skill>=1 and Skill<=4),
    PRIMARY KEY (Syear, ep_no, Stage_No),
    FOREIGN KEY (Syear, Ep_No) REFERENCES Episode(Syear, Ep_no) on
delete cascade,
    FOREIGN KEY (Song_ID) REFERENCES Song(Number) on delete cascade
);
```

21. StageIncludeTrainee (year, ep No, stage No, SSN trainee, role, no_of_votes)

```
-- StageIncludeTrSainee --
```

```
CREATE TABLE StageIncludeTrainee(
    SYear year NOT NULL,
    Ep_No INT(1) NOT NULL,
    Stage_No INT(3) NOT NULL,
    Ssn_trainee Numeric(12,0) NOT NULL,
    Role INT(1) NOT NULL DEFAULT 1,
    CHECK (Role>=1 and Role<=3),
    No_of_votes INT(3)
    CHECK (No_of_votes >= 0 AND No_of_votes <=500),
    PRIMARY KEY(SYear, Ep_No, Stage_No, Ssn_Trainee),
    FOREIGN KEY (SYear, Ep_No, Stage_No) references Stage(Syear,
Ep_No, Stage_no) on delete cascade,
    FOREIGN KEY (Ssn_trainee) REFERENCES Trainee(SsN) on delete
cascade
);
```

22. InvitedGuest (guest_ID)

```
-- InvitedGuest --
CREATE TABLE InvitedGuest(
    Guest_ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY
);
```

23. GGroup (Gname, no_of_member, guest_ID)

```
-- GGroup --
CREATE TABLE GGroup(
    Gname VARCHAR(100) NOT NULL PRIMARY KEY,
    No_of_member INT(2)
    CHECK (No_of_member >= 1 AND No_of_member <=20),
    Guest_ID INT NOT NULL,
    FOREIGN KEY (Guest_ID) references InvitedGuest(Guest_ID) ON
DELETE CASCADE
);
```

24. GroupSignatureSong (Gname, song_name)

```
-- GroupSignatureSong --
CREATE TABLE GroupSignatureSong(
    Gname VARCHAR(100) NOT NULL,
    Song_name VARCHAR(100) NOT NULL,
    PRIMARY KEY (Gname, Song_name),
    FOREIGN KEY (Gname) References GGroup(Gname) ON DELETE CASCADE
);
```

25. GuestSupportStage (guest_ID, year, ep No, stage No)

```
-- GuestSupportStage --
CREATE TABLE GuestSupportStage(
    Guest_ID INT NOT NULL,
    Syear year not nULL,
    Ep_No INT(1) NOT NULL,
    Stage_No INT(3) NOT NULL,
    PRIMARY KEY (SYear, Ep_No, Stage_No),
    FOREIGN KEY (Guest_ID) REFERENCES InvitedGuest(Guest_ID) ON
DELETE CASCADE,
    FOREIGN KEY (Syear, Ep_No, Stage_No) REFERENCES Stage(Syear,
Ep_No, Stage_No) ON DELETE CASCADE
);
```

II. Insert

Insert data for all tables in the database.

Requirements: The data in the tables must be meaningful, and each table has at least 4 rows (except table 14 – at least 2 rows).

-- Company --

```
INSERT INTO COMPANY VALUES
('C100',"Ghost","Dien      Bien      Phu      Street,      District      10,
HCMC",08361869275,'1992-08-19'),
('C101',"Ancient","Nguyen    Van    Khoi    Street,    District    Go    Vap,
HCMC",01285102784,'1978-03-24'),
('C102',"Deviant","Ly      Thai      To      Street,      District      10,
HCMC",09768256719,'1984-06-13'),
('C103',"Asgard","Bui      Thi      Xuan      Street,      District      1,
HCMC",03958269132,'2000-11-05');
```

-- Person --

```
INSERT INTO Person VALUES
(111120180001,"Johnny","Dang","Dong Van Cong Street, District 2, HCMC",0354887693),
(111120180002,"Luon","Vuituoi","Cong Quynh Street, District 1, HCMC",0354887694),
(111120180003,"Leo","Nguyen","Ly Thuong Kiet Street, District 10, HCMC",0354887695),
(111120180004,"Daniel","Wellington","Ky Con Street, District 1, HCMC",0354887696),
(111120180005,"Julliet","Johnson","Tran Hung Dao Street, District 1, HCMC",0354887697),
(111120180006,"Romeo","Smith","Tran Dinh Xu Street, District 1, HCMC",0354887698),
(111120180007,"Walter","White","Le Loi Street, District 1, HCMC",0354887699),
(111120180008,"Don","Brown","Nguyen Hue Street, District 1, HCMC",0354887700),
(111120180009,"Boba","Green","Tran Hung Dao Street, District 5, HCMC",0354887701),
(111120180010,"Samael","Morning","Ho Hao Hon Street, District 1, HCMC",0354887702),
(111120180011,"Lucifer","Morningstar","Le Lai Street, District 1, HCMC",0354887703),
(111120180012,"Don","Chisao","Nguyen Du Street, District 1, HCMC",0354887704),
(111120180013,"Tom","Vecna","Tran Hung Dao Street, District 1, HCMC",0354887705),
(111120180014,"Jerry","Peterson","Tran Dinh Xu Street, District 1, HCMC",0354887706),
(111120180015,"Sam","Waltersom","Le Loi Street, District 1, HCMC",0354887707),
(111120180016,"Ching","Sunat","Nguyen Hue Street, District 1, HCMC",0354887708),
(111120180017,"Sinat","San","Tran Hung Dao Street, District 5, HCMC",0354887709),
(111120180018,"Mohamale","Christ","Ho Hao Hon Street, District 1, HCMC",0354887710),
(111120180019,"John","Smith","Le Lai Street, District 1, HCMC",0354887711),
(111120180020,"Dat","Tran","Nguyen Du Street, District 1, HCMC",0354887712),
(111120180021,"Khoa","Nguyen","Dong Van Cong Street, District 2, HCMC",0354887713),
(111120180022,"Minh","Tran","Cong Quynh Street, District 1, HCMC",0354887714),
(111120180023,"Thien","Ly","Ly Thuong Kiet Street, District 10, HCMC",0354887715),
(111120180024,"Nguyen","Le","Ky Con Street, District 1, HCMC",0354887716),
(111120180025,"Nghia","Dinh","Tran Hung Dao Street, District 1, HCMC",0354887717),
(111120180026,"Hao","Mai","Tran Dinh Xu Street, District 1, HCMC",0354887718),
```

(111120180027,"Ron","Wesley","Le Loi Street, District 1, HCMC",0354887719),
(111120180028,"Emma","Watson","Nguyen Hue Street, District 1, HCMC",0354887720),
(111120180029,"Harry","Stone","Tran Hung Dao Street, District 5, HCMC",0354887721),
(111120180030,"Charles","Tree","Ho Hao Hon Street, District 1, HCMC",0354887722),
(111120180031,"Sam","Leaf","Dong Van Cong Street, District 2, HCMC",0354888822),
(111120180032,"Ching","Ly","Cong Quynh Street, District 1, HCMC",0354888823),
(111120180033,"Sinat","Pham","Ly Thuong Kiet Street, District 10, HCMC",0354888824),
(111120180034,"Mohamale","John","Ky Con Street, District 1, HCMC",0354888825),
(111120180035,"John","Robert","Tran Hung Dao Street, District 1, HCMC",0354888826),
(111120180036,"Dat","Michael","Tran Dinh Xu Street, District 1, HCMC",0354888827),
(111120180037,"Khoa","William","Le Loi Street, District 1, HCMC",0354888828),
(111120180038,"Minh","David","Ly Thuong Kiet Street, District 10, HCMC",0354888829),
(111120180039,"Thien","Richard","Ky Con Street, District 1, HCMC",0354888830),
(111120180040,"Nguyen","Joseph","Tran Hung Dao Street, District 1, HCMC",0354888831),
(111120190001,"Ella","Ice","Le Lai Street, District 1, HCMC",0354887723),
(111120190002,"Elizabeth","Queenie","Nguyen Du Street, District 1, HCMC",0354887724),
(111120190003,"Jane","Root","Tran Hung Dao Street, District 1, HCMC",0354887725),
(111120190004,"Ngan","Roof","Tran Dinh Xu Street, District 1, HCMC",0354887726),
(111120190005,"Thuy","Phan","Le Loi Street, District 1, HCMC",0354887727),
(111120190006,"Duy","Tham","Nguyen Hue Street, District 1, HCMC",0354887728),
(111120190007,"Tri","Leaf","Tran Hung Dao Street, District 5, HCMC",0354887729),
(111120190008,"Hieu","Ly","Ho Hao Hon Street, District 1, HCMC",0354887730),
(111120190009,"Hoa","Pham","Le Lai Street, District 1, HCMC",0354887731),
(111120190010,"Teo","John","Dong Van Cong Street, District 2, HCMC",0354887732),
(111120190011,"Ti","Robert","Cong Quynh Street, District 1, HCMC",0354887733),
(111120190012,"Tan","Michael","Ly Thuong Kiet Street, District 10, HCMC",0354887734),
(111120190013,"Tai","William","Ky Con Street, District 1, HCMC",0354887735),
(111120190014,"Chuong","David","Tran Hung Dao Street, District 1, HCMC",0354887736),
(111120190015,"Vy","Richard","Tran Dinh Xu Street, District 1, HCMC",0354887737),
(111120190016,"Nam","Joseph","Le Loi Street, District 1, HCMC",0354887738),
(111120190017,"Quan","Charles","Nguyen Hue Street, District 1, HCMC",0354887739),
(111120190018,"Sang","Thomas","Tran Hung Dao Street, District 5, HCMC",0354887740),
(111120190019,"Chieu","Ti","Ho Hao Hon Street, District 1, HCMC",0354887741),
(111120190020,"Toi","Tan","Le Lai Street, District 1, HCMC",0354887742),
(111120190021,"Khuya","Tai","Nguyen Du Street, District 1, HCMC",0354887743),
(111120190022,"Jacob","Chuong","Tran Hung Dao Street, District 1, HCMC",0354887744),
(111120190023,"Oliver","Jake","Tran Dinh Xu Street, District 1, HCMC",0354887745),
(111120190024,"John","Jack","Le Loi Street, District 1, HCMC",0354887746),
(111120190025,"Robert","Harry","Nguyen Hue Street, District 1, HCMC",0354887747),
(111120190026,"Michael","Jacob","Tran Hung Dao Street, District 5, HCMC",0354887748),
(111120190027,"William","Charlie","Ho Hao Hon Street, District 1, HCMC",0354887749),
(111120190028,"David","Thomas","Le Lai Street, District 1, HCMC",0354887750),
(111120190029,"Richard","George","Nguyen Du Street, District 1, HCMC",0354887751),
(111120190030,"Joseph","Oscar","Dong Van Cong Street, District 2, HCMC",0354887752),
(111120190031,"Oliver","Jake","Ho Hao Hon Street, District 1, HCMC",0354888853),
(111120190032,"John","Jack","Le Loi Street, District 1, HCMC",0354888854),

```
(111120190033,"Robert","Harry","Tran Hung Dao Street, District 1, HCMC",0354888855),
(111120190034,"Michael","Jacob","Tran Dinh Xu Street, District 1, HCMC",0354888856),
(111120190035,"Ti","Charlie","Le Loi Street, District 1, HCMC",0354888857),
(111120190036,"Tan","Thomas","Nguyen Hue Street, District 1, HCMC",0354888858),
(111120190037,"Tai","George","Tran Dinh Xu Street, District 1, HCMC",0354888859),
(111120190038,"Charles","Chieu","Le Loi Street, District 1, HCMC",0354888860),
(111120190039,"Thomas","Toi","Nguyen Hue Street, District 1, HCMC",0354888861),
(111120190040,"Ti","Khuya","Le Lai Street, District 1, HCMC",0354888862),
(222220180001,"Charles","James","Cong Quynh Street, District 1, HCMC",0354887753),
(222220180002,"Thomas","William","Ly Thuong Kiet Street, District 10, HCMC",0354887754),
(222220180003,"Margaret","Emma","Le Lai Street, District 1, HCMC",0354887755),
(222220180004,"Olivia","Samantha","Nguyen Du Street, District 1, HCMC",0354887756),
(222220190001,"Isla","Bethany","Tran Hung Dao Street, District 1, HCMC",0354887757),
(222220190002,"Emily","Elizabeth","Tran Dinh Xu Street, District 1, HCMC",0354887758),
(222220190003,"Poppy","Joanne","Le Loi Street, District 1, HCMC",0354887759),
(222220190004,"Ava","Megan","Nguyen Hue Street, District 1, HCMC",0354887760),
(333320180001,"Patricia","Elizabeth","Tran Hung Dao Street, District 5,
HCMC",0354887761),
(333320190001,"Jennifer","Linda","Ho Hao Hon Street, District 1, HCMC",0354887762);
```

```
-- Trainee --
```

```
INSERT INTO Trainee VALUES
```

```
(111120180001,'1998-02-11',"uploads/sheaf-of-rice.png","C100"),
(111120180002,'1999-10-23',"uploads/sheaf-of-rice.png","C101"),
(111120180003,'1985-05-07',"uploads/sheaf-of-rice.png","C102"),
(111120180004,'2002-10-07',"uploads/sheaf-of-rice.png","C103"),
(111120180005,'1997-02-11',"uploads/sheaf-of-rice.png","NULL"),
(111120180006,'1996-03-12',"uploads/sheaf-of-rice.png","C100"),
(111120180007,'1995-04-11',"uploads/sheaf-of-rice.png","C101"),
(111120180008,'1994-05-10',"uploads/sheaf-of-rice.png","C102"),
(111120180009,'1993-06-08',"uploads/sheaf-of-rice.png","C103"),
(111120180010,'1992-07-07',"uploads/sheaf-of-rice.png","NULL"),
(111120180011,'1991-08-06',"uploads/sheaf-of-rice.png","C100"),
(111120180012,'1990-09-04',"uploads/sheaf-of-rice.png","C101"),
(111120180013,'1989-10-03',"uploads/sheaf-of-rice.png","C102"),
(111120180014,'1988-11-01',"uploads/sheaf-of-rice.png","C103"),
(111120180015,'1987-12-01',"uploads/sheaf-of-rice.png","NULL"),
(111120180016,'1986-12-30',"uploads/sheaf-of-rice.png","C100"),
(111120180017,'1998-02-11',"uploads/sheaf-of-rice.png","C101"),
(111120180018,'1999-10-23',"uploads/sheaf-of-rice.png","C102"),
(111120180019,'1985-05-07',"uploads/sheaf-of-rice.png","C103"),
```

```
(111120180020, '2002-10-07', "uploads/sheaf-of-rice.png", "NULL"),
(111120180021, '1997-02-11', "uploads/sheaf-of-rice.png", "C100"),
(111120180022, '1996-03-12', "uploads/sheaf-of-rice.png", "C101"),
(111120180023, '1995-04-11', "uploads/sheaf-of-rice.png", "C102"),
(111120180024, '1994-05-10', "uploads/sheaf-of-rice.png", "C103"),
(111120180025, '1993-06-08', "uploads/sheaf-of-rice.png", "NULL"),
(111120180026, '1992-07-07', "uploads/sheaf-of-rice.png", "C100"),
(111120180027, '1991-08-06', "uploads/sheaf-of-rice.png", "C101"),
(111120180028, '1990-09-04', "uploads/sheaf-of-rice.png", "C102"),
(111120180029, '1989-10-03', "uploads/sheaf-of-rice.png", "C103"),
(111120180030, '1988-11-01', "uploads/sheaf-of-rice.png", "NULL"),
(111120180031, '1989-10-04', "uploads/sheaf-of-rice.png", "C100"),
(111120180032, '1988-11-02', "uploads/sheaf-of-rice.png", "C101"),
(111120180033, '1989-10-05', "uploads/sheaf-of-rice.png", "C102"),
(111120180034, '1988-11-03', "uploads/sheaf-of-rice.png", "C103"),
(111120180035, '1989-10-06', "uploads/sheaf-of-rice.png", "NULL"),
(111120180036, '1988-11-04', "uploads/sheaf-of-rice.png", "C100"),
(111120180037, '1989-10-07', "uploads/sheaf-of-rice.png", "C101"),
(111120180038, '1988-11-05', "uploads/sheaf-of-rice.png", "C102"),
(111120180039, '1989-10-08', "uploads/sheaf-of-rice.png", "C103"),
(111120180040, '1988-11-06', "uploads/sheaf-of-rice.png", "NULL"),
(111120190001, '1987-12-01', "uploads/sheaf-of-rice.png", "C100"),
(111120190002, '1986-12-30', "uploads/sheaf-of-rice.png", "C101"),
(111120190003, '1998-02-11', "uploads/sheaf-of-rice.png", "C102"),
(111120190004, '1999-10-23', "uploads/sheaf-of-rice.png", "C103"),
(111120190005, '1985-05-07', "uploads/sheaf-of-rice.png", "NULL"),
(111120190006, '2002-10-07', "uploads/sheaf-of-rice.png", "C100"),
(111120190007, '1997-02-11', "uploads/sheaf-of-rice.png", "C101"),
(111120190008, '1996-03-12', "uploads/sheaf-of-rice.png", "C102"),
(111120190009, '1995-04-11', "uploads/sheaf-of-rice.png", "C103"),
(111120190010, '1994-05-10', "uploads/sheaf-of-rice.png", "NULL"),
(111120190011, '1993-06-08', "uploads/sheaf-of-rice.png", "C100"),
(111120190012, '1992-07-07', "uploads/sheaf-of-rice.png", "C101"),
(111120190013, '1991-08-06', "uploads/sheaf-of-rice.png", "C102"),
(111120190014, '1990-09-04', "uploads/sheaf-of-rice.png", "C103"),
(111120190015, '1989-10-03', "uploads/sheaf-of-rice.png", "NULL"),
(111120190016, '1988-11-01', "uploads/sheaf-of-rice.png", "C100"),
```

```
(111120190017, '1987-12-01', "uploads/sheaf-of-rice.png", "C101"),
(111120190018, '1986-12-30', "uploads/sheaf-of-rice.png", "C102"),
(111120190019, '1998-02-11', "uploads/sheaf-of-rice.png", "C103"),
(111120190020, '1999-10-23', "uploads/sheaf-of-rice.png", "NULL"),
(111120190021, '1985-05-07', "uploads/sheaf-of-rice.png", "C100"),
(111120190022, '2002-10-07', "uploads/sheaf-of-rice.png", "C101"),
(111120190023, '1997-02-11', "uploads/sheaf-of-rice.png", "C102"),
(111120190024, '1996-03-12', "uploads/sheaf-of-rice.png", "C103"),
(111120190025, '1995-04-11', "uploads/sheaf-of-rice.png", "NULL"),
(111120190026, '1994-05-10', "uploads/sheaf-of-rice.png", "C100"),
(111120190027, '1993-06-08', "uploads/sheaf-of-rice.png", "C101"),
(111120190028, '1992-07-07', "uploads/sheaf-of-rice.png", "C102"),
(111120190029, '1991-08-06', "uploads/sheaf-of-rice.png", "C103"),
(111120190030, '1990-09-04', "uploads/sheaf-of-rice.png", "NULL"),
(111120190031, '1991-08-07', "uploads/sheaf-of-rice.png", "C100"),
(111120190032, '1990-09-05', "uploads/sheaf-of-rice.png", "C101"),
(111120190033, '1991-08-08', "uploads/sheaf-of-rice.png", "C102"),
(111120190034, '1990-09-06', "uploads/sheaf-of-rice.png", "C103"),
(111120190035, '1991-08-09', "uploads/sheaf-of-rice.png", "NULL"),
(111120190036, '1990-09-07', "uploads/sheaf-of-rice.png", "C100"),
(111120190037, '1991-08-10', "uploads/sheaf-of-rice.png", "C101"),
(111120190038, '1990-09-08', "uploads/sheaf-of-rice.png", "C102"),
(111120190039, '1991-08-11', "uploads/sheaf-of-rice.png", "C103"),
(111120190040, '1990-09-09', "uploads/sheaf-of-rice.png", "NULL");
```

```
-- MC --
```

```
INSERT INTO MC VALUES
```

```
(333320180001), (333320190001), (111120180022), (111120180023);
```

```
-- Mentor --
```

```
INSERT INTO Mentor VALUES
```

```
(222220180001), (222220180002), (222220180003), (222220180004),
(222220190001), (222220190002), (222220190003), (222220190004);
```

```
-- Song --
```

```
INSERT INTO Song (Released_year, Name, Singer_SSN_first_performed )
VALUES
```

```
(2018, 'Themesong 2018', NULL), (2019, 'Themesong 2019', NULL),
(2018, 'Perfect', NULL), (2020, 'Intentions', 222220180001),
(2010, 'Misery', 222220180002), (2015, 'Lean On', 222220180003),
(1999, 'Unstoppable', NULL), (2000, 'Data', NULL),
(2001, 'Sensitive', NULL), (2002, 'Happy birthday to you', NULL);

-- Themesong --
INSERT INTO ThemeSong VALUES ('S001'), ('S002');

-- SongComposedBy --
INSERT INTO SongComposedBy VALUES
('S001', 222220190001), ('S002', 222220190002), ('S003', 222220190003),
('S004', 222220190001), ('S005', 222220190002), ('S006', 222220190003);

-- Singer --
INSERT INTO Singer VALUES (222220180001, 1), (222220180002, 2), (222220180003, NULL);

-- SingerSignatureSong --
INSERT INTO SingerSignatureSong VALUES
(222220180001, 'Hello'), (222220180002, 'I believe I can fly'),
(222220180003, 'The Earth');

-- Producer --
INSERT INTO Producer VALUES (222220180004), (222220190004);

-- ProducerProgram --
INSERT INTO ProducerProgram VALUES
(222220180004, 'LoveYou3000'), (222220180004, 'Dark Pictures'),
(222220190004, 'Mementos'), (222220190004, 'Persona ');

-- SongWriter --
INSERT INTO SongWriter VALUES
(222220190001), (222220190002), (222220190003);

-- Season --
INSERT INTO Season VALUES

(2018, 'Hanoi', 'S001', 333320180001), (2019, 'HCMC', 'S002', 333320190001);
```



```
-- SeasonMentor --
```

```
INSERT INTO SeasonMentor VALUES
```

```
('2018',222220180001), ('2018',222220180002), ('2018',222220180003),  
('2018',222220180004), ('2019',222220190001), ('2019',222220190002),  
('2019',222220190003), ('2019',222220190004);
```

```
-- SeansonTrainee --
```

```
INSERT INTO SeasonTrainee VALUES
```

```
('2018',111120180001), ('2018',111120180002), ('2018',111120180003),  
('2018',111120180004), ('2018',111120180005), ('2018',111120180006),  
('2018',111120180007), ('2018',111120180008), ('2018',111120180009),  
('2018',111120180010), ('2018',111120180011), ('2018',111120180012),  
('2018',111120180013), ('2018',111120180014), ('2018',111120180015),  
('2018',111120180016), ('2018',111120180017), ('2018',111120180018),  
('2018',111120180019), ('2018',111120180020), ('2018',111120180021),  
('2018',111120180022), ('2018',111120180023), ('2018',111120180024),  
('2018',111120180025), ('2018',111120180026), ('2018',111120180027),  
('2018',111120180028), ('2018',111120180029), ('2018',111120180030),  
('2018',111120180031), ('2018',111120180032), ('2018',111120180033),  
('2018',111120180034), ('2018',111120180035), ('2018',111120180036),  
('2018',111120180037), ('2018',111120180038), ('2018',111120180039),  
('2018',111120180040), ('2019',111120190001), ('2019',111120190002),  
('2019',111120190003), ('2019',111120190004), ('2019',111120190005),  
('2019',111120190006), ('2019',111120190007), ('2019',111120190008),  
('2019',111120190009), ('2019',111120190010), ('2019',111120190011),  
('2019',111120190012), ('2019',111120190013), ('2019',111120190014),  
('2019',111120190015), ('2019',111120190016), ('2019',111120190017),  
('2019',111120190018), ('2019',111120190019), ('2019',111120190020),  
('2019',111120190021), ('2019',111120190022), ('2019',111120190023),  
('2019',111120190024), ('2019',111120190025), ('2019',111120190026),  
('2019',111120190027), ('2019',111120190028), ('2019',111120190029),  
('2019',111120190030), ('2019',111120190031), ('2019',111120190032),  
('2019',111120190033), ('2019',111120190034), ('2019',111120190035),  
('2019',111120190036), ('2019',111120190037), ('2019',111120190038),  
('2019',111120190039), ('2019',111120190040), ('2019',111120180003),  
('2019',111120180006), ('2019',111120180009), ('2019',111120180012),
```

```
('2019',111120180015), ('2019',111120180018), ('2019',111120180021),
('2019',111120180024), ('2019',111120180027), ('2019',111120180030),
('2019',111120180033), ('2019',111120180036), ('2019',111120180039);
```

```
-- MentorValuateTrainee --
```

```
INSERT INTO MentorValuateTrainee VALUES
```

```
('2018',111120180001,222220180001,20), ('2018',111120180002,222220180001,5),
('2018',111120180003,222220180001,78), ('2018',111120180004,222220180001,21),
('2018',111120180005,222220180001,43), ('2018',111120180006,222220180001,70),
('2018',111120180007,222220180001,21), ('2018',111120180008,222220180001,6),
('2018',111120180009,222220180001,61), ('2018',111120180010,222220180001,95),
('2018',111120180011,222220180001,12), ('2018',111120180012,222220180001,83),
('2018',111120180013,222220180001,37), ('2018',111120180014,222220180001,12),
('2018',111120180015,222220180001,89), ('2018',111120180016,222220180001,31),
('2018',111120180017,222220180001,99), ('2018',111120180018,222220180001,35),
('2018',111120180019,222220180001,65), ('2018',111120180020,222220180001,1),
('2018',111120180021,222220180001,46), ('2018',111120180022,222220180001,48),
('2018',111120180023,222220180001,78), ('2018',111120180024,222220180001,6),
('2018',111120180025,222220180001,63), ('2018',111120180026,222220180001,60),
('2018',111120180027,222220180001,96), ('2018',111120180028,222220180001,82),
('2018',111120180029,222220180001,2), ('2018',111120180030,222220180001,11),
('2018',111120180031,222220180001,0), ('2018',111120180032,222220180001,0),
('2018',111120180033,222220180001,0), ('2018',111120180034,222220180001,0),
('2018',111120180035,222220180001,0), ('2018',111120180036,222220180001,0),
('2018',111120180037,222220180001,0), ('2018',111120180038,222220180001,0),
('2018',111120180039,222220180001,0), ('2018',111120180040,222220180001,0),
('2018',111120180001,222220180002,73), ('2018',111120180002,222220180002,21),
('2018',111120180003,222220180002,14), ('2018',111120180004,222220180002,46),
('2018',111120180005,222220180002,30), ('2018',111120180006,222220180002,64),
('2018',111120180007,222220180002,47), ('2018',111120180008,222220180002,4),
('2018',111120180009,222220180002,32), ('2018',111120180010,222220180002,21),
('2018',111120180011,222220180002,9), ('2018',111120180012,222220180002,2),
('2018',111120180013,222220180002,74), ('2018',111120180014,222220180002,18),
('2018',111120180015,222220180002,81), ('2018',111120180016,222220180002,46),
('2018',111120180017,222220180002,9), ('2018',111120180018,222220180002,80),
('2018',111120180019,222220180002,38), ('2018',111120180020,222220180002,90),
('2018',111120180021,222220180002,53), ('2018',111120180022,222220180002,20),
('2018',111120180023,222220180002,73), ('2018',111120180024,222220180002,42),
('2018',111120180025,222220180002,66), ('2018',111120180026,222220180002,52),
('2018',111120180027,222220180002,46), ('2018',111120180028,222220180002,27),
```

('2018',111120180029,222220180002,68), ('2018',111120180030,222220180002,12),
('2018',111120180031,222220180002,0), ('2018',111120180032,222220180002,0),
('2018',111120180033,222220180002,0), ('2018',111120180034,222220180002,0),
('2018',111120180035,222220180002,0), ('2018',111120180036,222220180002,0),
('2018',111120180037,222220180002,0), ('2018',111120180038,222220180002,0),
('2018',111120180039,222220180002,0), ('2018',111120180040,222220180002,0),
('2018',111120180001,222220180003,91), ('2018',111120180002,222220180003,17),
('2018',111120180003,222220180003,33), ('2018',111120180004,222220180003,94),
('2018',111120180005,222220180003,48), ('2018',111120180006,222220180003,11),
('2018',111120180007,222220180003,55), ('2018',111120180008,222220180003,68),
('2018',111120180009,222220180003,44), ('2018',111120180010,222220180003,67),
('2018',111120180011,222220180003,81), ('2018',111120180012,222220180003,19),
('2018',111120180013,222220180003,98), ('2018',111120180014,222220180003,58),
('2018',111120180015,222220180003,89), ('2018',111120180016,222220180003,35),
('2018',111120180017,222220180003,39), ('2018',111120180018,222220180003,29),
('2018',111120180019,222220180003,2), ('2018',111120180020,222220180003,53),
('2018',111120180021,222220180003,72), ('2018',111120180022,222220180003,15),
('2018',111120180023,222220180003,92), ('2018',111120180024,222220180003,41),
('2018',111120180025,222220180003,0), ('2018',111120180026,222220180003,72),
('2018',111120180027,222220180003,6), ('2018',111120180028,222220180003,76),
('2018',111120180029,222220180003,82), ('2018',111120180030,222220180003,52),
('2018',111120180031,222220180003,0), ('2018',111120180032,222220180003,0),
('2018',111120180033,222220180003,0), ('2018',111120180034,222220180003,0),
('2018',111120180035,222220180003,0), ('2018',111120180036,222220180003,0),
('2018',111120180037,222220180003,0), ('2018',111120180038,222220180003,0),
('2018',111120180039,222220180003,0), ('2018',111120180040,222220180003,0),
('2018',111120180001,222220180004,10), ('2018',111120180002,222220180004,46),
('2018',111120180003,222220180004,1), ('2018',111120180004,222220180004,58),
('2018',111120180005,222220180004,87), ('2018',111120180006,222220180004,85),
('2018',111120180007,222220180004,28), ('2018',111120180008,222220180004,84),
('2018',111120180009,222220180004,83), ('2018',111120180010,222220180004,7),
('2018',111120180011,222220180004,3), ('2018',111120180012,222220180004,34),
('2018',111120180013,222220180004,18), ('2018',111120180014,222220180004,20),
('2018',111120180015,222220180004,49), ('2018',111120180016,222220180004,98),
('2018',111120180017,222220180004,13), ('2018',111120180018,222220180004,95),
('2018',111120180019,222220180004,4), ('2018',111120180020,222220180004,49),
('2018',111120180021,222220180004,77), ('2018',111120180022,222220180004,12),
('2018',111120180023,222220180004,35), ('2018',111120180024,222220180004,11),
('2018',111120180025,222220180004,24), ('2018',111120180026,222220180004,43),
('2018',111120180027,222220180004,49), ('2018',111120180028,222220180004,95),
('2018',111120180029,222220180004,5), ('2018',111120180030,222220180004,70),

('2018',111120180031,222220180004,0), ('2018',111120180032,222220180004,0),
('2018',111120180033,222220180004,0), ('2018',111120180034,222220180004,0),
('2018',111120180035,222220180004,0), ('2018',111120180036,222220180004,0),
('2018',111120180037,222220180004,0), ('2018',111120180038,222220180004,0),
('2018',111120180039,222220180004,0), ('2018',111120180040,222220180004,0),
('2019',111120190001,222220190001,55), ('2019',111120190002,222220190001,74),
('2019',111120190003,222220190001,66), ('2019',111120190004,222220190001,11),
('2019',111120190005,222220190001,93), ('2019',111120190006,222220190001,35),
('2019',111120190007,222220190001,57), ('2019',111120190008,222220190001,8),
('2019',111120190009,222220190001,44), ('2019',111120190010,222220190001,18),
('2019',111120190011,222220190001,68), ('2019',111120190012,222220190001,69),
('2019',111120190013,222220190001,9), ('2019',111120190014,222220190001,9),
('2019',111120190015,222220190001,74), ('2019',111120190016,222220190001,98),
('2019',111120190017,222220190001,81), ('2019',111120190018,222220190001,6),
('2019',111120190019,222220190001,64), ('2019',111120190020,222220190001,77),
('2019',111120190021,222220190001,60), ('2019',111120190022,222220190001,12),
('2019',111120190023,222220190001,24), ('2019',111120190024,222220190001,98),
('2019',111120190025,222220190001,32), ('2019',111120190026,222220190001,81),
('2019',111120190027,222220190001,72), ('2019',111120190028,222220190001,98),
('2019',111120190029,222220190001,65), ('2019',111120190030,222220190001,44),
('2019',111120190031,222220190001,88), ('2019',111120190032,222220190001,19),
('2019',111120190033,222220190001,79), ('2019',111120190034,222220190001,70),
('2019',111120190035,222220190001,60), ('2019',111120190036,222220190001,80),
('2019',111120190037,222220190001,56), ('2019',111120190038,222220190001,50),
('2019',111120190039,222220190001,16), ('2019',111120190040,222220190001,54),
('2019',111120190001,222220190002,77), ('2019',111120190002,222220190002,11),
('2019',111120190003,222220190002,11), ('2019',111120190004,222220190002,47),
('2019',111120190005,222220190002,39), ('2019',111120190006,222220190002,94),
('2019',111120190007,222220190002,66), ('2019',111120190008,222220190002,21),
('2019',111120190009,222220190002,53), ('2019',111120190010,222220190002,68),
('2019',111120190011,222220190002,3), ('2019',111120190012,222220190002,36),
('2019',111120190013,222220190002,83), ('2019',111120190014,222220190002,54),
('2019',111120190015,222220190002,81), ('2019',111120190016,222220190002,46),
('2019',111120190017,222220190002,47), ('2019',111120190018,222220190002,72),
('2019',111120190019,222220190002,5), ('2019',111120190020,222220190002,64),
('2019',111120190021,222220190002,92), ('2019',111120190022,222220190002,37),
('2019',111120190023,222220190002,83), ('2019',111120190024,222220190002,15),
('2019',111120190025,222220190002,27), ('2019',111120190026,222220190002,67),
('2019',111120190027,222220190002,3), ('2019',111120190028,222220190002,79),
('2019',111120190029,222220190002,74), ('2019',111120190030,222220190002,35),
('2019',111120190031,222220190002,32), ('2019',111120190032,222220190002,55),

('2019',111120190033,222220190002,29), ('2019',111120190034,222220190002,16),
('2019',111120190035,222220190002,80), ('2019',111120190036,222220190002,52),
('2019',111120190037,222220190002,1), ('2019',111120190038,222220190002,67),
('2019',111120190039,222220190002,77), ('2019',111120190040,222220190002,2),
('2019',111120190001,222220190003,4), ('2019',111120190002,222220190003,97),
('2019',111120190003,222220190003,17), ('2019',111120190004,222220190003,17),
('2019',111120190005,222220190003,22), ('2019',111120190006,222220190003,78),
('2019',111120190007,222220190003,82), ('2019',111120190008,222220190003,79),
('2019',111120190009,222220190003,84), ('2019',111120190010,222220190003,2),
('2019',111120190011,222220190003,20), ('2019',111120190012,222220190003,30),
('2019',111120190013,222220190003,80), ('2019',111120190014,222220190003,94),
('2019',111120190015,222220190003,27), ('2019',111120190016,222220190003,97),
('2019',111120190017,222220190003,77), ('2019',111120190018,222220190003,24),
('2019',111120190019,222220190003,37), ('2019',111120190020,222220190003,52),
('2019',111120190021,222220190003,39), ('2019',111120190022,222220190003,44),
('2019',111120190023,222220190003,18), ('2019',111120190024,222220190003,37),
('2019',111120190025,222220190003,81), ('2019',111120190026,222220190003,87),
('2019',111120190027,222220190003,58), ('2019',111120190028,222220190003,53),
('2019',111120190029,222220190003,33), ('2019',111120190030,222220190003,10),
('2019',111120190031,222220190003,14), ('2019',111120190032,222220190003,91),
('2019',111120190033,222220190003,53), ('2019',111120190034,222220190003,40),
('2019',111120190035,222220190003,19), ('2019',111120190036,222220190003,83),
('2019',111120190037,222220190003,82), ('2019',111120190038,222220190003,57),
('2019',111120190039,222220190003,11), ('2019',111120190040,222220190003,51),
('2019',111120190001,222220190004,81), ('2019',111120190002,222220190004,65),
('2019',111120190003,222220190004,14), ('2019',111120190004,222220190004,72),
('2019',111120190005,222220190004,29), ('2019',111120190006,222220190004,57),
('2019',111120190007,222220190004,9), ('2019',111120190008,222220190004,99),
('2019',111120190009,222220190004,52), ('2019',111120190010,222220190004,15),
('2019',111120190011,222220190004,19), ('2019',111120190012,222220190004,61),
('2019',111120190013,222220190004,20), ('2019',111120190014,222220190004,12),
('2019',111120190015,222220190004,58), ('2019',111120190016,222220190004,19),
('2019',111120190017,222220190004,3), ('2019',111120190018,222220190004,11),
('2019',111120190019,222220190004,71), ('2019',111120190020,222220190004,53),
('2019',111120190021,222220190004,13), ('2019',111120190022,222220190004,40),
('2019',111120190023,222220190004,55), ('2019',111120190024,222220190004,39),
('2019',111120190025,222220190004,10), ('2019',111120190026,222220190004,77),
('2019',111120190027,222220190004,56), ('2019',111120190028,222220190004,51),
('2019',111120190029,222220190004,74), ('2019',111120190030,222220190004,71),
('2019',111120190031,222220190004,48), ('2019',111120190032,222220190004,71),
('2019',111120190033,222220190004,35), ('2019',111120190034,222220190004,51),

```
( '2019',111120190035,222220190004,5), ('2019',111120190036,222220190004,30),
('2019',111120190037,222220190004,29), ('2019',111120190038,222220190004,65),
('2019',111120190039,222220190004,69), ('2019',111120190040,222220190004,9),
('2019',111120180003,222220190001,54), ('2019',111120180006,222220190001,48),
('2019',111120180009,222220190001,88), ('2019',111120180012,222220190001,36),
('2019',111120180015,222220190001,68), ('2019',111120180018,222220190001,80),
('2019',111120180021,222220190001,82), ('2019',111120180024,222220190001,16),
('2019',111120180027,222220190001,26), ('2019',111120180030,222220190001,29),
('2019',111120180033,222220190001,60), ('2019',111120180036,222220190001,31),
('2019',111120180039,222220190001,93), ('2019',111120180003,222220190002,7),
('2019',111120180006,222220190002,43), ('2019',111120180009,222220190002,85),
('2019',111120180012,222220190002,82), ('2019',111120180015,222220190002,10),
('2019',111120180018,222220190002,22), ('2019',111120180021,222220190002,18),
('2019',111120180024,222220190002,88), ('2019',111120180027,222220190002,79),
('2019',111120180030,222220190002,98), ('2019',111120180033,222220190002,67),
('2019',111120180036,222220190002,28), ('2019',111120180039,222220190002,26),
('2019',111120180003,222220190003,67), ('2019',111120180006,222220190003,18),
('2019',111120180009,222220190003,58), ('2019',111120180012,222220190003,11),
('2019',111120180015,222220190003,62), ('2019',111120180018,222220190003,15),
('2019',111120180021,222220190003,19), ('2019',111120180024,222220190003,26),
('2019',111120180027,222220190003,31), ('2019',111120180030,222220190003,74),
('2019',111120180033,222220190003,33), ('2019',111120180036,222220190003,31),
('2019',111120180039,222220190003,59), ('2019',111120180003,222220190004,66),
('2019',111120180006,222220190004,22), ('2019',111120180009,222220190004,59),
('2019',111120180012,222220190004,1), ('2019',111120180015,222220190004,15),
('2019',111120180018,222220190004,51), ('2019',111120180021,222220190004,20),
('2019',111120180024,222220190004,17), ('2019',111120180027,222220190004,55),
('2019',111120180030,222220190004,83), ('2019',111120180033,222220190004,87),
('2019',111120180036,222220190004,5), ('2019',111120180039,222220190004,5);
```

```
-- Episode --
```

```
INSERT INTO Episode VALUES
```

```
('2018',1,"2018_ep1",'2018-01-11 19:00',90),
('2018',2,"2018_ep2",'2018-02-12 19:00',90),
('2018',3,"2018_ep3",'2018-03-16 19:00',90),
('2018',4,"2018_ep4",'2018-04-17 19:00',90),
('2018',5,"2018_ep5",'2018-05-19 19:00',90),
('2019',1,"2019_ep1",'2019-07-05 19:00',90),
('2019',2,"2019_ep2",'2019-08-06 19:00',90),
('2019',3,"2019_ep3",'2019-09-07 19:00',90),
('2019',4,"2019_ep4",'2019-10-09 19:00',90),
```

```

('2019',5,"2019_ep5",'2019-11-10 19:00',90);

-- Stage --
INSERT INTO Stage VALUES
('2018',1,801,0,4,NULL,"S001"), ('2018',2,821,1,1,0,"S003"),
('2018',2,822,1,1,0,"S004"), ('2018',2,823,1,2,0,"S005"),
('2018',2,824,1,2,0,"S006"), ('2018',2,825,1,3,0,"S007"),
('2018',2,826,1,3,0,"S008"), ('2018',3,831,1,4,0,"S009"),
('2018',3,832,1,4,0,"S009"), ('2018',3,833,1,4,0,"S010"),
('2018',3,834,1,4,0,"S010"), ('2018',4,841,1,4,0,"S005"),
('2018',4,842,1,4,0,"S006"), ('2018',4,843,1,4,0,"S007"),
('2018',4,844,1,4,0,"S008"), ('2018',5,850,0,1,NULL,"S006"),
('2018',5,851,0,3,NULL,"S007"), ('2018',5,852,0,4,NULL,"S008"),
('2018',5,853,0,2,NULL,"S009"), ('2018',5,854,0,1,NULL,"S009"),
('2018',5,855,0,3,NULL,"S010"), ('2018',5,856,0,4,NULL,"S010"),
('2018',5,857,0,3,NULL,"S005"), ('2018',5,858,0,2,NULL,"S006"),
('2018',5,859,0,3,NULL,"S007"), ('2018',5,860,1,4,NULL,"S008"),
('2018',5,861,1,4,NULL,"S009"), ('2019',1,901,0,4,NULL,"S002"),
('2019',2,921,1,1,0,"S006"), ('2019',2,922,1,2,0,"S007"),
('2019',2,923,1,3,0,"S008"), ('2019',2,924,1,1,0,"S009"),
('2019',2,925,1,2,0,"S009"), ('2019',2,926,1,3,0,"S010"),
('2019',3,931,1,4,0,"S006"), ('2019',3,932,1,4,0,"S009"),
('2019',3,933,1,4,0,"S009"), ('2019',3,934,1,4,0,"S006"),
('2019',4,941,1,1,0,"S005"), ('2019',4,942,1,3,0,"S006"),
('2019',4,943,1,4,0,"S007"), ('2019',4,944,1,1,0,"S008"),
('2019',5,950,0,2,NULL,"S006"), ('2019',5,951,0,3,NULL,"S007"),
('2019',5,952,0,4,NULL,"S008"), ('2019',5,953,0,1,NULL,"S009"),
('2019',5,954,0,2,NULL,"S009"), ('2019',5,955,0,3,NULL,"S010"),
('2019',5,956,0,1,NULL,"S010"), ('2019',5,957,0,4,NULL,"S005"),
('2019',5,958,0,2,NULL,"S006"), ('2019',5,959,0,1,NULL,"S007"),
('2019',5,960,1,2,NULL,"S008"), ('2019',5,961,1,3,NULL,"S009");

-- -- StageIncludeTrainee --
INSERT INTO StageIncludeTrainee VALUES
('2018',1,801,111120180001,1,NULL), ('2018',1,801,111120180002,1,NULL),
('2018',1,801,111120180003,1,NULL), ('2018',1,801,111120180004,1,NULL),
('2018',1,801,111120180005,1,NULL), ('2018',1,801,111120180006,1,NULL),
('2018',1,801,111120180007,1,NULL), ('2018',1,801,111120180008,1,NULL),

```

('2018',1,801,111120180009,1,NULL), ('2018',1,801,111120180010,1,NULL),
('2018',1,801,111120180011,1,NULL), ('2018',1,801,111120180012,1,NULL),
('2018',1,801,111120180013,1,NULL), ('2018',1,801,111120180014,1,NULL),
('2018',1,801,111120180015,1,NULL), ('2018',1,801,111120180016,1,NULL),
('2018',1,801,111120180017,1,NULL), ('2018',1,801,111120180018,1,NULL),
('2018',1,801,111120180019,1,NULL), ('2018',1,801,111120180020,1,NULL),
('2018',1,801,111120180021,1,NULL), ('2018',1,801,111120180022,1,NULL),
('2018',1,801,111120180023,1,NULL), ('2018',1,801,111120180024,1,NULL),
('2018',1,801,111120180025,1,NULL), ('2018',1,801,111120180026,1,NULL),
('2018',1,801,111120180027,1,NULL), ('2018',1,801,111120180028,1,NULL),
('2018',1,801,111120180029,1,NULL), ('2018',1,801,111120180030,1,NULL),
('2018',1,801,111120180031,1,NULL), ('2018',1,801,111120180032,1,NULL),
('2018',1,801,111120180033,1,NULL), ('2018',1,801,111120180034,1,NULL),
('2018',1,801,111120180035,1,NULL), ('2018',1,801,111120180036,1,NULL),
('2018',1,801,111120180037,1,NULL), ('2018',1,801,111120180038,1,NULL),
('2018',1,801,111120180039,1,NULL), ('2018',1,801,111120180040,1,NULL),
('2018',2,821,111120180001,2,360), ('2018',2,821,111120180002,3,65),
('2018',2,821,111120180003,1,1), ('2018',2,821,111120180004,1,497),
('2018',2,821,111120180005,1,420), ('2018',2,822,111120180006,2,1),
('2018',2,822,111120180007,3,383), ('2018',2,822,111120180008,1,33),
('2018',2,822,111120180009,1,1), ('2018',2,822,111120180010,1,47),
('2018',2,823,111120180011,2,50), ('2018',2,823,111120180012,3,1),
('2018',2,823,111120180013,1,461), ('2018',2,823,111120180014,1,486),
('2018',2,823,111120180015,1,1), ('2018',2,824,111120180016,2,297),
('2018',2,824,111120180017,3,432), ('2018',2,824,111120180018,1,1),
('2018',2,824,111120180019,1,52), ('2018',2,824,111120180020,1,371),
('2018',2,825,111120180021,2,1), ('2018',2,825,111120180022,3,358),
('2018',2,825,111120180023,1,47), ('2018',2,825,111120180024,1,1),
('2018',2,825,111120180025,1,156), ('2018',2,826,111120180026,2,245),
('2018',2,826,111120180027,3,1), ('2018',2,826,111120180028,1,91),
('2018',2,826,111120180029,1,464), ('2018',2,826,111120180030,1,1),
('2018',3,831,111120180001,2,331), ('2018',3,831,111120180002,3,470),
('2018',3,831,111120180004,1,424), ('2018',3,831,111120180005,1,308),
('2018',3,831,111120180007,1,423), ('2018',3,832,111120180008,2,183),
('2018',3,832,111120180010,3,120), ('2018',3,832,111120180011,1,150),
('2018',3,832,111120180013,1,100), ('2018',3,832,111120180014,1,50),
('2018',3,833,111120180016,2,274), ('2018',3,833,111120180017,3,279),

('2018',3,833,111120180019,1,454), ('2018',3,833,111120180020,1,399),
('2018',3,833,111120180022,1,261), ('2018',3,834,111120180023,2,184),
('2018',3,834,111120180025,3,151), ('2018',3,834,111120180026,1,198),
('2018',3,834,111120180028,1,130), ('2018',3,834,111120180029,1,126),
('2018',4,841,111120180001,2,360), ('2018',4,841,111120180002,3,198),
('2018',4,841,111120180004,1,455), ('2018',4,841,111120180005,1,72),
('2018',4,842,111120180007,2,351), ('2018',4,842,111120180008,3,472),
('2018',4,842,111120180010,1,137), ('2018',4,842,111120180011,1,101),
('2018',4,843,111120180016,2,426), ('2018',4,843,111120180017,3,455),
('2018',4,843,111120180019,1,26), ('2018',4,843,111120180020,1,84),
('2018',4,844,111120180022,2,151), ('2018',4,844,111120180023,3,342),
('2018',4,844,111120180025,1,98), ('2018',4,844,111120180026,1,55),
('2018',5,850,111120180001,1,419), ('2018',5,851,111120180002,1,437),
('2018',5,852,111120180004,1,402), ('2018',5,853,111120180007,1,445),
('2018',5,854,111120180008,1,149), ('2018',5,855,111120180010,1,400),
('2018',5,856,111120180016,1,251), ('2018',5,857,111120180017,1,70),
('2018',5,858,111120180022,1,13), ('2018',5,859,111120180023,1,277),
('2018',5,860,111120180001,2,NULL), ('2018',5,860,111120180002,3,NULL),
('2018',5,860,111120180004,1,NULL), ('2018',5,860,111120180007,1,NULL),
('2018',5,860,111120180008,1,NULL), ('2018',5,861,111120180010,2,NULL),
('2018',5,861,111120180016,3,NULL), ('2018',5,861,111120180017,1,NULL),
('2018',5,861,111120180022,1,NULL), ('2018',5,861,111120180023,1,NULL),
('2019',1,901,111120190001,1,NULL), ('2019',1,901,111120190002,1,NULL),
('2019',1,901,111120190003,1,NULL), ('2019',1,901,111120190004,1,NULL),
('2019',1,901,111120190005,1,NULL), ('2019',1,901,111120190006,1,NULL),
('2019',1,901,111120190007,1,NULL), ('2019',1,901,111120190008,1,NULL),
('2019',1,901,111120190009,1,NULL), ('2019',1,901,111120190010,1,NULL),
('2019',1,901,111120190011,1,NULL), ('2019',1,901,111120190012,1,NULL),
('2019',1,901,111120190013,1,NULL), ('2019',1,901,111120190014,1,NULL),
('2019',1,901,111120190015,1,NULL), ('2019',1,901,111120190016,1,NULL),
('2019',1,901,111120190017,1,NULL), ('2019',1,901,111120190018,1,NULL),
('2019',1,901,111120190019,1,NULL), ('2019',1,901,111120190020,1,NULL),
('2019',1,901,111120190021,1,NULL), ('2019',1,901,111120190022,1,NULL),
('2019',1,901,111120190023,1,NULL), ('2019',1,901,111120190024,1,NULL),
('2019',1,901,111120190025,1,NULL), ('2019',1,901,111120190026,1,NULL),
('2019',1,901,111120190027,1,NULL), ('2019',1,901,111120190028,1,NULL),

```

('2019',1,901,111120190029,1,NULL), ('2019',1,901,111120190030,1,NULL),
('2019',1,901,111120190031,1,NULL), ('2019',1,901,111120190032,1,NULL),
('2019',1,901,111120190033,1,NULL), ('2019',1,901,111120190034,1,NULL),
('2019',1,901,111120190035,1,NULL), ('2019',1,901,111120190036,1,NULL),
('2019',1,901,111120190037,1,NULL), ('2019',1,901,111120190038,1,NULL),
('2019',1,901,111120190039,1,NULL), ('2019',1,901,111120190040,1,NULL),
('2019',1,901,111120180003,1,NULL), ('2019',1,901,111120180006,1,NULL),
('2019',1,901,111120180009,1,NULL), ('2019',1,901,111120180012,1,NULL),
('2019',1,901,111120180015,1,NULL), ('2019',1,901,111120180018,1,NULL),
('2019',1,901,111120180021,1,NULL), ('2019',1,901,111120180024,1,NULL),
('2019',1,901,111120180027,1,NULL), ('2019',1,901,111120180030,1,NULL),
('2019',1,901,111120180033,1,NULL), ('2019',1,901,111120180036,1,NULL),
('2019',1,901,111120180039,1,NULL), ('2019',2,921,111120190026,2,121),
('2019',2,921,111120180009,3,31), ('2019',2,921,111120180030,1,14),
('2019',2,921,111120190028,1,93), ('2019',2,921,111120190006,1,5),
('2019',2,922,111120190016,2,445), ('2019',2,922,111120190002,3,94),
('2019',2,922,111120180033,1,6), ('2019',2,922,111120190029,1,183),
('2019',2,922,111120190020,1,261), ('2019',2,923,111120190036,2,69),
('2019',2,923,111120190015,3,404), ('2019',2,923,111120190038,1,160),
('2019',2,923,111120190032,1,383), ('2019',2,923,111120190009,1,94),
('2019',2,924,111120190001,2,462), ('2019',2,924,111120190007,3,341),
('2019',2,924,111120190017,1,209), ('2019',2,924,111120190008,1,151),
('2019',2,924,111120190021,1,353), ('2019',2,925,111120190033,2,278),
('2019',2,925,111120190012,3,282), ('2019',2,925,111120180003,1,98),
('2019',2,925,111120190013,1,483), ('2019',2,925,111120180027,1,303),
('2019',2,926,111120190024,2,449), ('2019',2,926,111120190027,3,315),
('2019',2,926,111120180039,1,157), ('2019',2,926,111120190005,1,46),
('2019',2,926,111120190031,1,190), ('2019',3,931,111120190013,2,432),
('2019',3,931,111120190001,3,375), ('2019',3,931,111120190024,1,126),
('2019',3,931,111120190016,1,486), ('2019',3,931,111120190015,1,39),
('2019',3,932,111120190032,2,179), ('2019',3,932,111120190021,3,396),
('2019',3,932,111120190007,1,219), ('2019',3,932,111120190027,1,79),
('2019',3,932,111120180027,1,16), ('2019',3,933,111120190012,2,274),
('2019',3,933,111120190033,3,195), ('2019',3,933,111120190020,1,134),
('2019',3,933,111120190017,1,190), ('2019',3,933,111120190031,1,359),
('2019',3,934,111120190029,2,51), ('2019',3,934,111120190038,3,169),
('2019',3,934,111120180039,1,133), ('2019',3,934,111120190008,1,11),

```

```
( '2019', 3, 934, 111120190026, 1, 220 ), ( '2019', 4, 941, 111120190015, 2, 352 ),
( '2019', 4, 941, 111120190024, 3, 355 ), ( '2019', 4, 941, 111120180039, 1, 455 ),
( '2019', 4, 941, 111120190020, 1, 408 ), ( '2019', 4, 942, 111120190038, 2, 304 ),
( '2019', 4, 942, 111120190032, 3, 88 ), ( '2019', 4, 942, 111120190017, 1, 445 ),
( '2019', 4, 942, 111120190033, 1, 209 ), ( '2019', 4, 943, 111120190007, 2, 202 ),
( '2019', 4, 943, 111120190026, 3, 43 ), ( '2019', 4, 943, 111120190012, 1, 56 ),
( '2019', 4, 943, 111120190031, 1, 258 ), ( '2019', 4, 944, 111120190001, 2, 87 ),
( '2019', 4, 944, 111120190021, 3, 420 ), ( '2019', 4, 944, 111120190013, 1, 333 ),
( '2019', 4, 944, 111120190016, 1, 166 ), ( '2019', 5, 950, 111120180039, 1, 123 ),
( '2019', 5, 951, 111120190017, 1, 320 ), ( '2019', 5, 952, 111120190021, 1, 388 ),
( '2019', 5, 953, 111120190020, 1, 125 ), ( '2019', 5, 954, 111120190024, 1, 481 ),
( '2019', 5, 955, 111120190015, 1, 126 ), ( '2019', 5, 956, 111120190013, 1, 89 ),
( '2019', 5, 957, 111120190038, 1, 116 ), ( '2019', 5, 958, 111120190031, 1, 251 ),
( '2019', 5, 959, 111120190033, 1, 43 ), ( '2019', 5, 960, 111120180039, 2, NULL ),
( '2019', 5, 960, 111120190017, 3, NULL ), ( '2019', 5, 960, 111120190021, 1, NULL ),
( '2019', 5, 960, 111120190020, 1, NULL ), ( '2019', 5, 960, 111120190024, 1, NULL ),
( '2019', 5, 961, 111120190015, 2, NULL ), ( '2019', 5, 961, 111120190013, 3, NULL ),
( '2019', 5, 961, 111120190038, 1, NULL ), ( '2019', 5, 961, 111120190031, 1, NULL ),
( '2019', 5, 961, 111120190033, 1, NULL );
```

```
-- InvitedGuest --
```

```
INSERT INTO InvitedGuest VALUES (1), (2), (3), (4), (5), (6), (7), (8);
```

```
-- Group --
```

```
INSERT INTO GGroup VALUES
```

```
('MenInBlack', 5, 5), ('Beatles', 6, 6), ('Walkers', 4, 7), ('Cters', 5, 8);
```

```
-- GroupSignatureSong --
```

```
INSERT INTO GroupSignatureSong VALUES
```

```
('MenInBlack', 'See You Again'), ('Beatles', 'Hello'),
```

```
('Walkers', 'Faded'), ('Cters', 'Despacito');
```

```
-- GuestSupportStage --
```

```
INSERT INTO GuestSupportStage VALUES
```

```
(2, 2018, 4, 841), (4, 2018, 4, 842), (6, 2018, 4, 843), (8, 2018, 4, 844),
```

```
(1, 2019, 4, 941), (3, 2019, 4, 942), (5, 2019, 4, 943), (7, 2019, 4, 944);
```

PART 2: STORE PROCEDURE, FUNCTION, TRIGGER (3 points)**I. Trigger****1.1 Trigger to check a trainee can participate 3 seasons at most****a. Create trigger**

```
-- 1. Trigger to check constraint --
-- a. Trainee can participate 3 seasons at most --
DELIMITER //
CREATE OR REPLACE TRIGGER at_most_three_season
AFTER INSERT ON seasontrainee
FOR EACH ROW
BEGIN
    DECLARE Total_year int;
    SELECT count(s.Syear) INTO Total_year FROM seasontrainee s
WHERE s.Ssn_trainee = new.Ssn_trainee GROUP BY s.Ssn_trainee;
    IF (Total_year > 3) THEN
        SIGNAL SQLSTATE '45000' SET
            MESSAGE_TEXT = 'Max 3 season';
    END IF;
END;
//
DELIMITER ;

DELIMITER //
CREATE OR REPLACE TRIGGER at_most_three_season_update
AFTER UPDATE ON seasontrainee
FOR EACH ROW
BEGIN
    DECLARE Total_year int;
    SELECT count(s.Syear) INTO Total_year FROM seasontrainee s
WHERE s.Ssn_trainee = new.Ssn_trainee GROUP BY s.Ssn_trainee;
    IF (Total_year > 3) THEN
        SIGNAL SQLSTATE '45000' SET
            MESSAGE_TEXT = 'Max 3 season';
    END IF;
END;
//
DELIMITER ;
```

b. Test trigger

b1. Insert

First, we insert two more seasons: 2020 and 2021 in “season” table.

Suppose that we choose the trainee with SSN = 111120180005 taking part in season 2018.

Run these statements below (Choose SSN of trainee who did not participate in the debut night of previous season).

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES
('2019', '111120180005')
```

✓ 1 row inserted.

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES ('2019', '111120180005');
```

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES
('2020', '111120180005')
```

✓ 1 row inserted.

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES ('2020', '111120180005');
```

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES
('2021', '111120180005')
```

Since the trainee has joined 3 season, the error textbox is shown below:

Error

SQL query: [Copy](#) [Edit](#)

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES ('2021', '111120180005')
```

MySQL said: ?

#1644 - Max 3 season

1.2 Trigger to check a trainee success to join in a debut night, then he/she cannot register as a trainee of later seasons

a. Create trigger

-- b. If a trainee success to join in a debut night, then he/she cannot register as a trainee of later seasons --

DELIMITER //

CREATE OR REPLACE TRIGGER join_debut_night

AFTER INSERT ON seasontrainee

FOR EACH ROW

BEGIN

 DECLARE Debut_season year;

 DECLARE errorMessage text;

 SELECT s.SYear INTO Debut_season FROM stageincludetrainee s

 WHERE s.Ep_No = 5 and s.Ssn_trainee = new.Ssn_trainee

 ORDER BY s.SYear DESC

 LIMIT 1;

 IF (debut_season IS NOT NULL) THEN

 IF (new.Syear > debut_season) THEN

 SET errorMessage = concat("Joined debut night on ",
debut_season);

 SIGNAL SQLSTATE '45000' SET

 MESSAGE_TEXT = errorMessage;

 END IF;

 END IF;

END;

//

DELIMITER ;

DELIMITER //

CREATE OR REPLACE TRIGGER join_debut_night_update

AFTER UPDATE ON seasontrainee

FOR EACH ROW

BEGIN

 DECLARE Debut_season year;

```
DECLARE errorMessage text;
SELECT s.SYear INTO Debut_season FROM stageincludetrainee s
WHERE s.Ep_No = 5 and s.Ssn_trainee = new.Ssn_trainee
ORDER BY s.SYear DESC
LIMIT 1;
IF (debut_season IS NOT NULL) THEN
    IF (new.Syear > debut_season) THEN
        SET errorMessage = concat("Joined debut night on ",
debut_season);
        SIGNAL SQLSTATE '45000' SET
            MESSAGE_TEXT = errorMessage;
    END IF;
END IF;
END;
```

//

DELIMITER ;


b. Test trigger

b1. Insert


```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES
('2019', '111120180001')
```

This trainee has joined in the debut night, Episode 5 of Season 2018, then he/she cannot register as a trainee in Season 2019.

Error

SQL query: [Copy](#)  [Edit](#)

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES ('2019', '111120180001')
```

MySQL said: 

#1644 - Joined debut night on 2018

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES
('2019', '111120180020');
```

Since he/she did not take part in debut night of Season 2018, the insert statement is accepted:

✓ 1 row inserted.

```
INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`) VALUES ('2019', '111120180020');
```

b2. Update

Suppose we update the Season year 2018 to 2019 of the trainee whose SSN = 111120180002

```
UPDATE `seasontrainee` SET `Syear` = '2019'
WHERE `seasontrainee`.`Syear` = 2018
AND CONCAT(`seasontrainee`.`Ssn_trainee`) = '111120180002'
```

Because he/she participated in debut night of Season 2018, we cannot update the value to other seasons.

Error

SQL query: [Copy](#)

```
UPDATE `seasontrainee` SET `Syear` = '2019'
WHERE `seasontrainee`.`Syear` = 2018
AND CONCAT(`seasontrainee`.`Ssn_trainee`) = '111120180002';
```

MySQL said: ?

#1644 - Joined debut night on 2018

We can update Season of trainee that did not join in debut night of any seasons:

```
UPDATE `seasontrainee` SET `Syear` = '2019'
WHERE `seasontrainee`.`Syear` = 2018
AND CONCAT(`seasontrainee`.`Ssn_trainee`) = '111120180005';
```

✓ 1 row affected. (Query took 0.0116 seconds.)

```
UPDATE `seasontrainee` SET `Syear` = '2019' WHERE `seasontrainee`.`Syear` = 2018
AND CONCAT(`seasontrainee`.`Ssn_trainee`) = '111120180005';
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

2. Trigger to calculate the total number of votes of a group (derived attribute of table 19) when number of votes of a member (no_of_votes of table 20) is updated

a. Create trigger

```
-- 2. Write trigger to calculate the total number of votes of a group
-- (derived attribute of table 19) --
-- when number of votes of a member (no_of_votes of table 20) is
updated --
DELIMITER //

CREATE OR REPLACE TRIGGER Update_total_vote
AFTER UPDATE ON stageincludetrainee
FOR EACH ROW
BEGIN
    DECLARE different int;
    SET different = new.No_of_votes - old.No_of_votes;
    UPDATE stage s
    SET s.Total_Votes = s.Total_Votes + different
    WHERE (s.Syear,s.Ep_No,s.Stage_No) = (new.Syear, new.Ep_no,
new.Stage_No);
END;

//
DELIMITER ;

DELIMITER //

CREATE OR REPLACE TRIGGER Insert_total_vote
AFTER INSERT ON stageincludetrainee
FOR EACH ROW
BEGIN
    DECLARE new_vote int;
    SET new_vote = new.No_of_votes;
    UPDATE stage s
    SET s.Total_Votes = s.Total_Votes + new_vote
    WHERE (s.Syear,s.Ep_No,s.Stage_No) = (new.Syear, new.Ep_no,
new.Stage_No);
END;
```

```
//
DELIMITER ;

DELIMITER //

CREATE OR REPLACE TRIGGER Delete_total_vote
AFTER DELETE ON stageincludetrainee
FOR EACH ROW
BEGIN
    DECLARE old_vote int;
    SET old_vote = old.No_of_votes;
    UPDATE stage s
    SET s.Total_Votes = s.Total_Votes - old_vote
    WHERE (s.Syear,s.Ep_No,s.Stage_No) = (old.Syear, old.Ep_no,
old.Stage_No);
END;

//
DELIMITER ;
```

b. Test trigger

b1. Update: The table below shows the total votes of Group who participated in 2018, Episode 2, Stage_No 821 is 1343 in “Stage” table

Syear	Ep_No	Stage_No	Is_Group	Skill	Total_Votes	Song_ID
2018	1	801	0	4	NULL	S001
2018	2	821	1	1	1343	S003
2018	2	822	1	1	465	S004

Now, in “stageincludetrainee” table, we update the value of the vote where SYear = 2018, Ep_No = 2, Stage_No = 821 and Ssn_trainee = 111120180003 to 50

```
UPDATE `stageincludetrainee` SET `No_of_votes` = '50'
WHERE `stageincludetrainee`.`SYear` = 2018
AND `stageincludetrainee`.`Ep_No` = 2
AND `stageincludetrainee`.`Stage_No` = 821
AND CONCAT(`stageincludetrainee`.`Ssn_trainee`) = '111120180003';
```

After updating, the total votes in “Stage” table has changed from 1343 to 1392

Year	Ep_No	Stage_No	Is_Group	Skill	Total_Votes	Song_ID
2018	1	801	0	4	NULL	S001
2018	2	821	1	1	1392	S003
2018	2	822	1	1	465	S004

b2. Delete

Next, we delete the tuple of Ssn_trainee = 111120180003 in “stageincludetrainee” table

```
DELETE FROM stageincludetrainee
```

```
WHERE `stageincludetrainee`.`SYear` = 2018
```

```
AND `stageincludetrainee`.`Ep_No` = 2
```

```
AND `stageincludetrainee`.`Stage_No` = 821
```

```
AND CONCAT(`stageincludetrainee`.`Ssn_trainee`) = '111120180003'
```

As a result, the total votes has changed from 1392 to 1342 in “stage” table

Year	Ep_No	Stage_No	Is_Group	Skill	Total_Votes	Song_ID
2018	1	801	0	4	NULL	S001
2018	2	821	1	1	1342	S003
2018	2	822	1	1	465	S004

b3. Insert

Lastly, we insert the tuple of Ssn_trainee = 111120180003 with total_votes = 126 in

“stageincludetrainee” table

```
INSERT INTO `stageincludetrainee`(`SYear`, `Ep_No`, `Stage_No`,  
`Ssn_trainee`, `Role`, `No_of_votes`) VALUES ('2018', '2', '821',  
'111120180003', '1', '126');
```

The total_votes has changed from 1342 to 1468 in “stage” table

Year	Ep_No	Stage_No	Is_Group	Skill	Total_Votes	Song_ID
2018	1	801	0	4	NULL	S001
2018	2	821	1	1	1468	S003
2018	2	822	1	1	465	S004

3.1 Trigger to ensure that a trainee has at most one group stage in episode 2, 3, 4

a. Create trigger

-- 3. Write trigger to ensure that a trainee has : --

-- a. at most one group stage in episode 2, 3, 4 --

DELIMITER //

CREATE OR REPLACE TRIGGER group_stage_check

AFTER INSERT ON stageincludetrainee

FOR EACH ROW

BEGIN

DECLARE group_stage_joined int(3);

DECLARE errorMessage text;

SELECT count(sit.Stage_No) INTO group_stage_joined

FROM stageincludetrainee sit join stage s

ON (sit.SYear, sit.Ep_No, sit.Stage_No) = (s.Syear, s.Ep_No, s.Stage_No)

WHERE sit.Syear = new.Syear AND sit.Ep_No = new.Ep_No AND new.Ep_no != 1 AND new.Ep_no
!= 5 AND s.Is_Group = 1 AND sit.Ssn_trainee = new.Ssn_trainee;

IF (group_stage_joined > 1) THEN

SET errorMessage = concat(new.Ssn_trainee, " already joined a group stage");

SIGNAL SQLSTATE '45000' SET

MESSAGE_TEXT = errorMessage;

END IF;

END;

//

DELIMITER ;

DELIMITER //

CREATE OR REPLACE TRIGGER group_stage_check_update

AFTER UPDATE ON stageincludetrainee

FOR EACH ROW

BEGIN

DECLARE group_stage_joined int(3);

DECLARE errorMessage text;

SELECT count(sit.Stage_No) INTO group_stage_joined

FROM stageincludetrainee sit join stage s

ON (sit.SYear, sit.Ep_No, sit.Stage_No) = (s.Syear, s.Ep_No, s.Stage_No)

WHERE sit.Syear = new.Syear AND sit.Ep_No = new.Ep_No AND new.Ep_no != 1 AND new.Ep_no
!= 5 AND s.Is_Group = 1 AND sit.Ssn_trainee = new.Ssn_trainee;

```
IF (group_stage_joined > 1) THEN
    SET errorMessage = concat(new.Ssn_trainee, " already joined a group stage");
    SIGNAL SQLSTATE '45000' SET
        MESSAGE_TEXT = errorMessage;
END IF;
END;
//
DELIMITER ;
```

b. Test trigger

b1. Insert

Assume that we choose the trainee with SSN = 111120180010 and Stage_No = 822.

Now we insert one more tuple of this trainee with Stage_No = 823.


```
INSERT INTO `stageincludetrainee` (`SYear`, `Ep_No`, `Stage_No`,
`Ssn_trainee`, `Role`, `No_of_votes`) VALUES ('2018', '2', '823',
'111120180010', '1', '47')
```

As he has joined the group stage whose Stage_No = 822, the error will occur and we get the result:

Error

SQL query: [Copy](#)

```
INSERT INTO `stageincludetrainee` (`SYear`, `Ep_No`, `Stage_No`, `Ssn_trainee`, `Role`, `No_
```

MySQL said: 

#1644 - 111120180010 already joined a group stage

b2. Update

We can change the value of Stage_No of any trainees since he/she can only join 1 group stage.

```
UPDATE `stageincludetrainee` SET `Stage_No` = '823'
WHERE `stageincludetrainee`.`SYear` = 2018
AND `stageincludetrainee`.`Ep_No` = 2
AND `stageincludetrainee`.`Stage_No` = 822
AND CONCAT(`stageincludetrainee`.`Ssn_trainee`) = '111120180010'
```

✓ 1 row affected. (Query took 0.0116 seconds.)

```
UPDATE `stageincludetrainee` SET `Stage_No` = '823' WHERE
`stageincludetrainee`.`SYear` = 2018 AND `stageincludetrainee`.`Ep_No` = 2 AND
`stageincludetrainee`.`Stage_No` = 822 AND
CONCAT(`stageincludetrainee`.`Ssn_trainee`) = '111120180010';
```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

3.2 Trigger to ensure that a trainee has at most one group stage and one individual stage in episode 5

a. Create trigger

-- b. at most one group stage and one individual stage in episode 5 --

```
DELIMITER //
CREATE OR REPLACE TRIGGER stage_five_check
AFTER INSERT ON stageincludetrainee
FOR EACH ROW
BEGIN
    DECLARE group_stage_joined int(3);
    DECLARE indi_stage_joined int(3);

    SELECT count(sit.Stage_No) INTO group_stage_joined
    FROM stageincludetrainee sit join stage s
    ON (sit.SYear, sit.Ep_No, sit.Stage_No) = (s.Syear, s.Ep_No, s.Stage_No)
    WHERE sit.Syear = new.Syear AND sit.Ep_no = 5 AND s.Is_Group = 1 AND sit.Ssn_trainee =
new.Ssn_trainee;

    SELECT count(sit.Stage_No) INTO indi_stage_joined
    FROM stageincludetrainee sit join stage s
    ON (sit.SYear, sit.Ep_No, sit.Stage_No) = (s.Syear, s.Ep_No, s.Stage_No)
    WHERE sit.Syear = new.Syear AND sit.Ep_no = 5 AND s.Is_Group = 0 AND sit.Ssn_trainee =
new.Ssn_trainee;

    IF (group_stage_joined > 1) THEN
        SIGNAL SQLSTATE '45000' SET
            MESSAGE_TEXT = "Already joined a group stage";
    END IF;

    IF (indi_stage_joined > 1) THEN
        SIGNAL SQLSTATE '45000' SET
            MESSAGE_TEXT = "Already joined a individual stage";
    END IF;

END;
```


```
//  
DELIMITER ;  
  
DELIMITER //  
CREATE OR REPLACE TRIGGER stage_five_check_update  
AFTER UPDATE ON stageincludetrainee  
FOR EACH ROW  
BEGIN  
    DECLARE group_stage_joined int(3);  
    DECLARE indi_stage_joined int(3);  
  
    SELECT count(sit.Stage_No) INTO group_stage_joined  
    FROM stageincludetrainee sit join stage s  
    ON (sit.SYear, sit.Ep_No, sit.Stage_No) = (s.Syear, s.Ep_No, s.Stage_No)  
    WHERE sit.Syear = new.Syear AND sit.Ep_no = 5 AND s.Is_Group = 1 AND sit.Ssn_trainee =  
new.Ssn_trainee;  
  
    SELECT count(sit.Stage_No) INTO indi_stage_joined  
    FROM stageincludetrainee sit join stage s  
    ON (sit.SYear, sit.Ep_No, sit.Stage_No) = (s.Syear, s.Ep_No, s.Stage_No)  
    WHERE sit.Syear = new.Syear AND sit.Ep_no = 5 AND s.Is_Group = 0 AND sit.Ssn_trainee =  
new.Ssn_trainee;  
  
    IF (group_stage_joined > 1) THEN  
        SIGNAL SQLSTATE '45000' SET  
            MESSAGE_TEXT = "Already joined a group stage";  
    END IF;  
  
    IF (indi_stage_joined > 1) THEN  
        SIGNAL SQLSTATE '45000' SET  
            MESSAGE_TEXT = "Already joined a individual stage";  
    END IF;  
  
END;  
  
//  
DELIMITER ;
```


b. Test trigger

```
INSERT INTO `stageincludetrainee` (`SYear`, `Ep_No`, `Stage_No`,  
`Ssn_trainee`, `Role`, `No_of_votes`) VALUES ('2018', '5', '851',  
'111120180001', '1', '419')
```

ErrorSQL query: [Copy](#)

```
INSERT INTO `stageincludetrainee` (`SYear`, `Ep_No`, `Stage_No`, `Ssn_trainee`, `Role`, `No_
```


MySQL said: 

#1644 - Already joined a individual stage

```
INSERT INTO `stageincludetrainee` (`SYear`, `Ep_No`, `Stage_No`,  
`Ssn_trainee`, `Role`, `No_of_votes`) VALUES ('2018', '5', '861',  
'111120180001', '1', '419')
```

ErrorSQL query: [Copy](#)

```
INSERT INTO `stageincludetrainee` (`SYear`, `Ep_No`, `Stage_No`, `Ssn_trainee`, `Role`, `No_
```

MySQL said: 

#1644 - Already joined a group stage

II. Store Procedure/Function

1. Procedure/function to print list of trainees come to the next episode (or debut if input episode is debut night)

- a. Create procedure/ function

```
-- II. Store Procedure/Function --
-- 1. Create a procedure/function to print list of trainees come to the
next episode (or debut if input episode is debut night) --
DELIMITER //
CREATE OR REPLACE PROCEDURE next_ep_list
(InYear year, InEp int(1))
BEGIN
    IF (InEp = 1) THEN
        SELECT s.Ssn_trainee, AVG(m.Score)
        FROM stageincludetrainee s JOIN mentorvaluatetrainee m
        ON s.SYear = m.Syear AND s.Ssn_trainee = m.Ssn_trainee
        WHERE (s.SYear, s.Ep_No) = (InYear, InEp)
        GROUP BY s.Ssn_trainee
        ORDER BY AVG(m.Score) DESC
        LIMIT 30;
    END IF;
    IF (InEp = 2) THEN
        SELECT s.Ssn_trainee, s.No_of_votes
        FROM stageincludetrainee s
        WHERE (s.SYear, s.Ep_No) = (InYear, InEp)
        ORDER BY s.No_of_votes DESC
        LIMIT 20;
    END IF;
    IF (InEp = 3) THEN
        SELECT sit.Ssn_trainee, sit.No_of_votes
        FROM stageincludetrainee sit
        WHERE (sit.SYear, sit.Ep_No) = (InYear, InEp)
        AND sit.Ssn_trainee NOT IN(
            SELECT lgt1.Ssn_trainee
            FROM(
                SELECT * FROM stageincludetrainee sit
                WHERE (sit.SYear, sit.Ep_No, sit.Stage_No) IN (
                    SELECT s.Syear, s.Ep_No, s.Stage_No
```

```

        FROM stage s
        WHERE (s.SYear, s.Ep_No) = (InYear, InEp)
        AND (s.Total_Votes) IN(
            SELECT MIN(s.Total_Votes)
            FROM stage s
            WHERE (s.SYear, s.Ep_No) = (InYear, InEp)
            GROUP BY s.Song_ID
        )
    )
) AS lgt1
WHERE (
    SELECT      COUNT(*)
    FROM(
        SELECT * FROM stageincludetrainee sit
        WHERE (sit.SYear, sit.Ep_No, sit.Stage_No) IN (
            SELECT s.Syear, s.Ep_No, s.Stage_No
            FROM stage s
            WHERE (s.SYear, s.Ep_No) = (InYear, InEp)
            AND (s.Total_Votes) IN(
                SELECT MIN(s.Total_Votes)
                FROM stage s
                WHERE (s.SYear, s.Ep_No) = (InYear, InEp)
                GROUP BY s.Song_ID
            )
        )
    ) AS lgt2
    WHERE lgt2.Stage_No = lgt1.Stage_No AND
          lgt2.No_of_votes <= lgt1.No_of_votes
    ) <= 2
)
ORDER BY sit.No_of_votes;
END IF;
IF (InEp = 4) THEN
    SELECT s.Ssn_trainee, s.No_of_votes
    FROM stageincludetrainee s
    WHERE (s.SYear, s.Ep_No) = (InYear, InEp)
    ORDER BY s.No_of_votes DESC

```

```
        LIMIT 10;
    END IF;
    IF (InEp = 5) THEN
        SELECT s.Ssn_trainee, s.No_of_votes
            FROM stageincludetrainee s
           WHERE (s.SYear, s.Ep_No) = (InYear, InEp)
          AND s.Stage_No IN(
                SELECT stage.Stage_No
                FROM stage
                WHERE (stage.Syear, stage.Ep_No, stage.Is_Group) = (InYear,
InEp, 0)
                )
           ORDER BY s.No_of_votes DESC
          LIMIT 5;
    END IF;
END;
//
DELIMITER ;
```

b. Call procedure/ function

Assume that we would like to check the list of trainees who will come to **Episode 4** in Season 2018, we can use the command:

```
CALL `next_ep_list` (2018,3);
```

We will obtain the result as the image:

✓ Showing rows 0 - 15 (16 total, Query took 0.0017 seconds.)

```
CALL `next_ep_list` (2018,3);
```

[[Edit inline](#)] [[Edit](#)] [[Create PHP code](#)]

☐ Show all | Number of rows: 25 ▾ Filter rows:

Extra options

Ssn_trainee	No_of_votes
111120180010	120
111120180011	150
111120180025	151
111120180008	183
111120180023	184
111120180026	198
111120180022	261
111120180016	274
111120180017	279
111120180005	308
111120180001	331
111120180020	399
111120180007	423
111120180004	424
111120180019	454
111120180002	470

2. Procedure/function to retrieve the result of a trainee in a season

a. Create procedure/ function

-- 2. Create a procedure/function to retrieve the result of a trainee in a season --

DELIMITER //

CREATE OR REPLACE PROCEDURE trainee_result

(InYear year, InSsn decimal(12,0))

BEGIN

 SELECT a.Ep_no as 'Episode', b.Score as 'Result'

 FROM(

 SELECT 1 as Ep_no, NULL as 'Score'

 UNION ALL

 SELECT 2 as Ep_no, NULL as 'Score'

 UNION ALL

 SELECT 3 as Ep_no, NULL as 'Score'

 UNION ALL

 SELECT 4 as Ep_no, NULL as 'Score'

 UNION ALL

 SELECT 5 as Ep_no, NULL as 'Score'

) AS a

 LEFT JOIN(

 SELECT 1 as Ep_no, format(AVG(mvt.Score),2) as 'Score'

 FROM mentorvaluatetrainee mvt

 WHERE (mvt.Syear, mvt.Ssn_trainee) = (InYear, InSsn)

 UNION

 SELECT 2 as Ep_no, format(sit.No_of_votes,0) as 'Score'

 FROM stageincludetrainee sit

 WHERE (sit.SYear, sit.Ep_No, sit.Ssn_trainee) = (InYear, 2,

InSsn)

 UNION

 SELECT 3 as Ep_no, format(sit.No_of_votes,0) as 'Score'

 FROM stageincludetrainee sit

```
        WHERE (sit.SYear, sit.Ep_No, sit.Ssn_trainee) = (InYear, 3,
InSsn)

    UNION

    SELECT 4 as Ep_no, format(sit.No_of_votes,0) as 'Score'
    FROM stageincludetrainee sit
        WHERE (sit.SYear, sit.Ep_No, sit.Ssn_trainee) = (InYear, 4,
InSsn)

    UNION

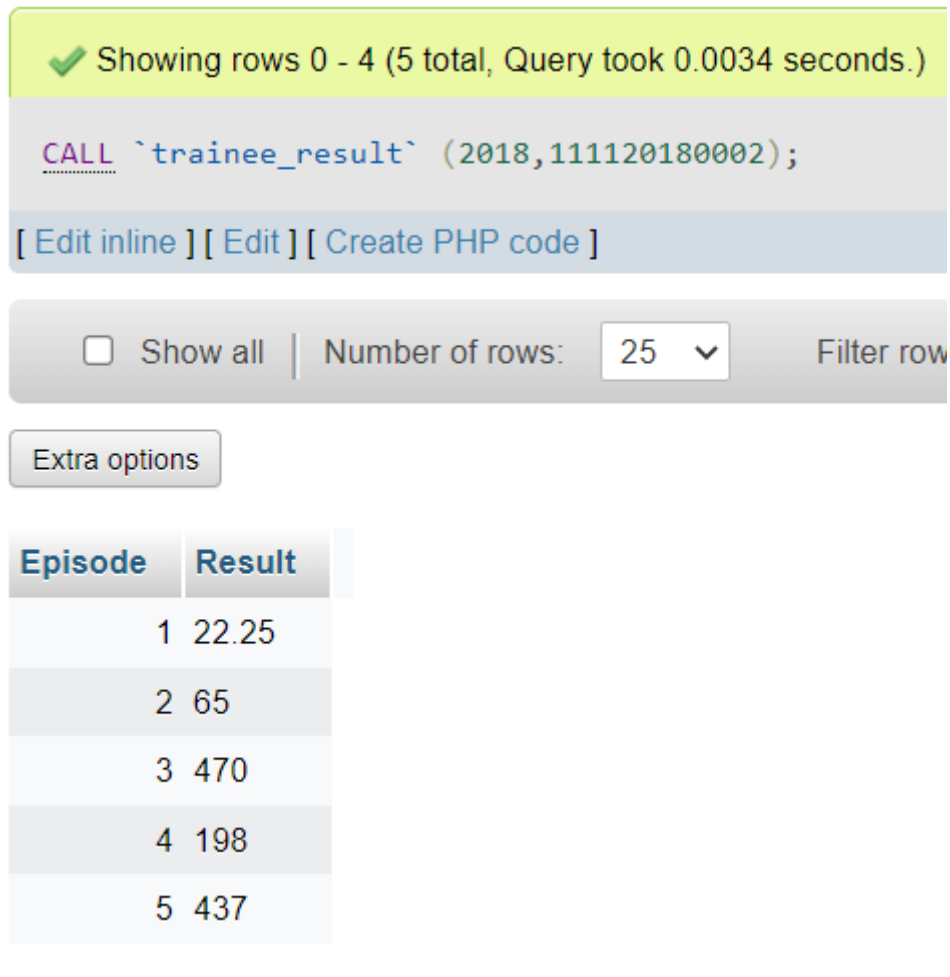
    SELECT 5 as Ep_no, format(sit.No_of_votes,0) as 'Score'
    FROM stageincludetrainee sit
    WHERE sit.Ssn_trainee = InSsn
    AND sit.Stage_No IN(
        SELECT s.Stage_No
        FROM stage s
        WHERE (s.Syear, s.Ep_No, s.Is_Group) = (InYear, 5, 0)
    )
    ) as b
    ON a.Ep_no = b.Ep_no;
END;
//
DELIMITER ;
```

b. Call procedure/ function

Assume that we would like to check the result of trainees who participate in 2018 and has the Ssn = 111120180002, we can use the following command:

```
CALL `trainee_result` (2018,111120180002);
```

We will obtain the result as the image:



✓ Showing rows 0 - 4 (5 total, Query took 0.0034 seconds.)

```
CALL `trainee_result` (2018,111120180002);
```

[Edit inline] [Edit] [Create PHP code]

☐ Show all | Number of rows: 25 ▼ Filter row

Extra options

Episode	Result
1	22.25
2	65
3	470
4	198
5	437

PART 3: BUILDING APPLICATIONS (3 points)**I. Create user****1. Create user**

```
CREATE USER 'sManager'@'localhost' IDENTIFIED BY 'password';
```

2. Grant privileges

```
GRANT ALL PRIVILEGES ON *.* TO `sManager`@`localhost` IDENTIFIED BY  
PASSWORD '*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19' WITH GRANT OPTION;
```

```
GRANT ALL PRIVILEGES ON `reality_show`.* TO `sManager`@`localhost`;
```

II. Programming environment

Apache, MySQL and PHP

III. How to connect application to MariDb (XAMPP) database

We use XAMPP to code our application. XAMPP is a free and Open-source cross-platform Web Server Solution Stack built by Apache Friends, which consists mostly of the Apache HTTP Server, MariaDB/MySQL Database, and interpreters for PHP and Perl scripts.

More detail will be provided in the online version (page 1)

IV. Implement features

1. Code for adding information for a new trainee

```

<?php
$servername = 'localhost';
$username = 'sManager';
$password = 'shop01';
$dbname = "reality_show";
try {
    $ssn = $_POST['ssn'];
    $fname = $_POST['fname'];
    $lname = $_POST['lname'];
    $phone = $_POST['phone'];
    $address = $_POST['address'];
    $dob = $_POST['dob'];
    $photo = 'uploads/sheaf-of-rice.png';
    $company = $_POST['company'];
    $year = $_POST['season'];
    $sql = "INSERT INTO `person` (Ssn,Fname,Lname,PAddress,Phone)
        VALUES ($ssn,'$fname','$lname','$address','$phone');
    INSERT INTO `trainee` (Ssn,DoB,Photo,Company_ID)
    VALUES ($ssn,'$dob','$photo','$company');
    INSERT INTO `seasontrainee` (`Syear`, `Ssn_trainee`)
    VALUES ('$year', '$ssn');";
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
    /* set the PDO error mode to exception */
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $conn->exec($sql);
    echo "New record created successfully";
} catch (PDOException $e) {

    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>

```

2. Code for searching information of trainees by name (full name of trainee includes the search string).

```

<?php
$where = "";
if (!empty($category_ids)):
    if (isset($_GET['search']) && !empty($_GET['search'])) {
        $where .= " and (p.Ssn LIKE '%".$_GET['search']."%' or p.Fname
        LIKE '%".$_GET['search']."%' or p.Lname LIKE

```

```

        '%{$_GET['search']}%' or (concat(p.Fname,' ',p.Lname) LIKE
        '%{$_GET['search']}%') )";
    }
    $trainee = $conn->query("SELECT DISTINCT t.*, p.Fname as fname, p.Lname
    as lname FROM `trainee` t inner join `seasontrainee` st on
    st.Ssn_trainee=t.Ssn inner join `season` s on st.Syear=s.SYear inner
    join person p on p.Ssn=t.Ssn where t.Ssn != 0 {$swhere} order by
    RAND()");
    while ($row = $trainee->fetch_assoc()):
        ?>

```

3. Code for getting trainee Ssn to show detail information

```

<?php
if (isset($_GET['Ssn']) && $_GET['Ssn'] > 0) {
    $qry = $conn->query("SELECT DISTINCT t.*, st.Syear as syear,
    p.Fname as fname, p.Lname as lname, p.PAddress as `address`, p.Phone as
    phone, count(st.Syear) as cyear FROM `trainee` t inner join
    `seasontrainee` st on st.Ssn_trainee=t.Ssn inner join `season` s on
    st.Syear=s.SYear inner join person p on p.Ssn=t.Ssn where t.Ssn != 0
    and t.Ssn = '{$_GET['Ssn']}'");
    if ($qry->num_rows > 0) {
        foreach ($qry->fetch_assoc() as $k => $v) {
            $$k = $v;
        }
    } else {
        echo "<script> alert('Unkown Ssn.');

```

4. Code for showing detail information

```

<div class="content py-3">
    <div class="card card-outline card-primary rounded-0 shadow">

```

```

<div class="card-header">
  <h2 class="card-title"><b>Trainee's detail infomation</b></h2>
</div>
<div class="card-body">
  <div class="container-fluid">
    <div id="msg"></div>
    <div class="row">
      <div class="col-lg-4 col-md-5 col-sm-12 text-center">
        <div class="position-relative overflow-hidden" id="prod-img-holder">
          " id="prod-img"
            class="img-thumbnail" style="background-color:#f2faf4">
        </div>
      </div>
      <div class="col-lg-8 col-md-7 col-sm-12">
        <h3><b>
          <?= $fname ?>
          <?= $lname ?> &nbsp;
        </b><i class="fas fa-user-tie fa-spin" style="color:#54c577"></i></h3>
        <div class="d-flex w-100" style="padding-bottom:10px">
          <div class="col-auto px-0">Ssn: &nbsp;</div>
          <h5 class="card-title text-truncate w-100 hah">
            <?= $Ssn ?> &nbsp;
          </h5>
        </div>
        <div class="d-flex" style="padding-bottom:10px">
          <div class="col-auto px-0">Date of birth: &nbsp;</div>
          <div class="col-auto px-0 flex-shrink-1 flex-grow-1">
            <p class="m-0">
              <?= $DoB ?>
            </p>
          </div>
        </div>
        <div class="d-flex" style="padding-bottom:10px">
          <div class="col-auto px-0">Company: &nbsp;</div>
          <div class="col-auto px-0 flex-shrink-1 flex-grow-1">
            <p class="m-0">

```



```
</div>
```

```
<?php
```

```
require_once('../config.php');
```

```
if (isset($_GET['year']) && isset($_GET['ssn'])) {
```

```
    $qry = $conn->query("CALL trainee_result('{$_GET['year']}','{$_GET['ssn']}') ");
```

```
    while ($row = $qry->fetch_assoc()) {
```

```
?>
```

```
<div class="container-fluid">
```

```
    <div class="row">
```

```
        <div class="col-3 border"><span class="font-weight-bolder">
```

```
            <?= isset($row['Episode']) ? $row['Episode'] : " ?>
```

```
        </span></div>
```

```
        <div class="col-3 border"><span class="font-weight-bolder">
```

```
            <?= isset($row['Result']) ? $row['Result'] : 'No result' ?>
```

```
        </span></div>
```

```
    </div>
```

```
</div>
```

```
<?php } ?>
```

```
<div class="clear-fix mb-3"></div>
```

```
<div class="text-right">
```

```
    <button class="btn btn-default bg-gradient-dark text-light btn-sm btn-flat" type="button"
```

```
data-dismiss="modal"><i
```

```
class="fa fa-times"></i> Close</button>
```

```
</div>
```

```
</div>
```

```
<?php
```

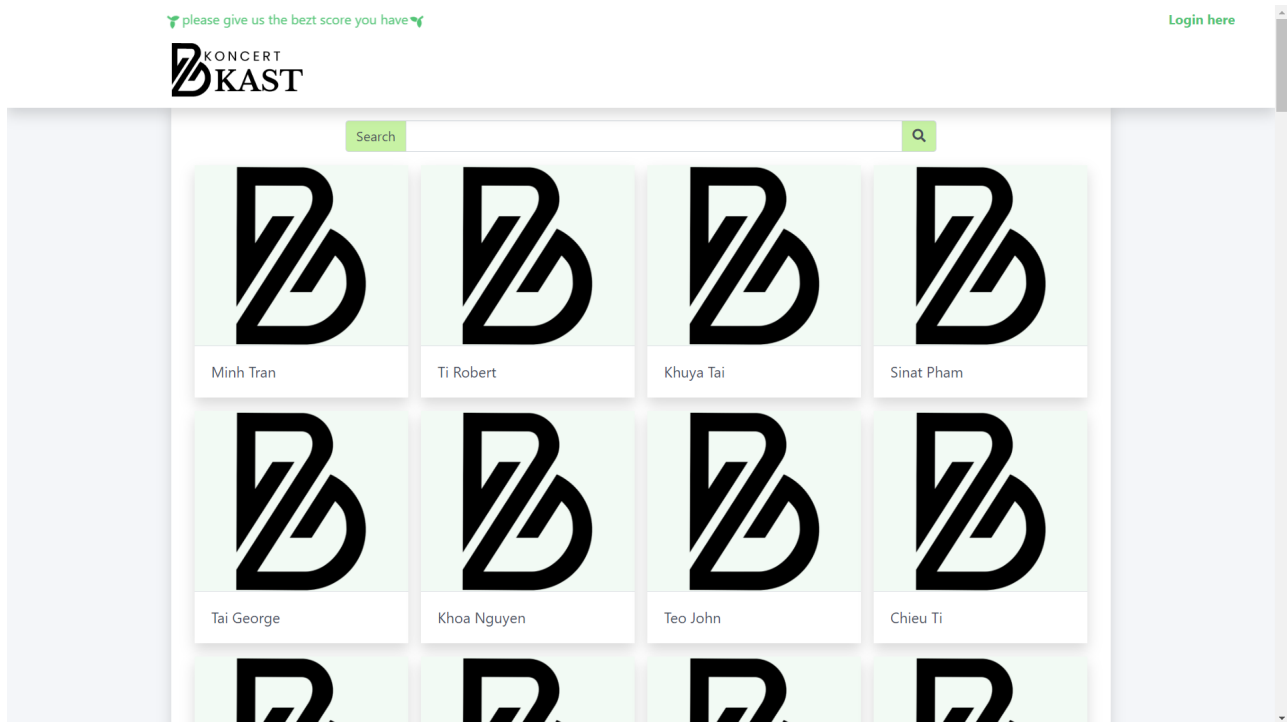
```
    exit;
```

```
}
```

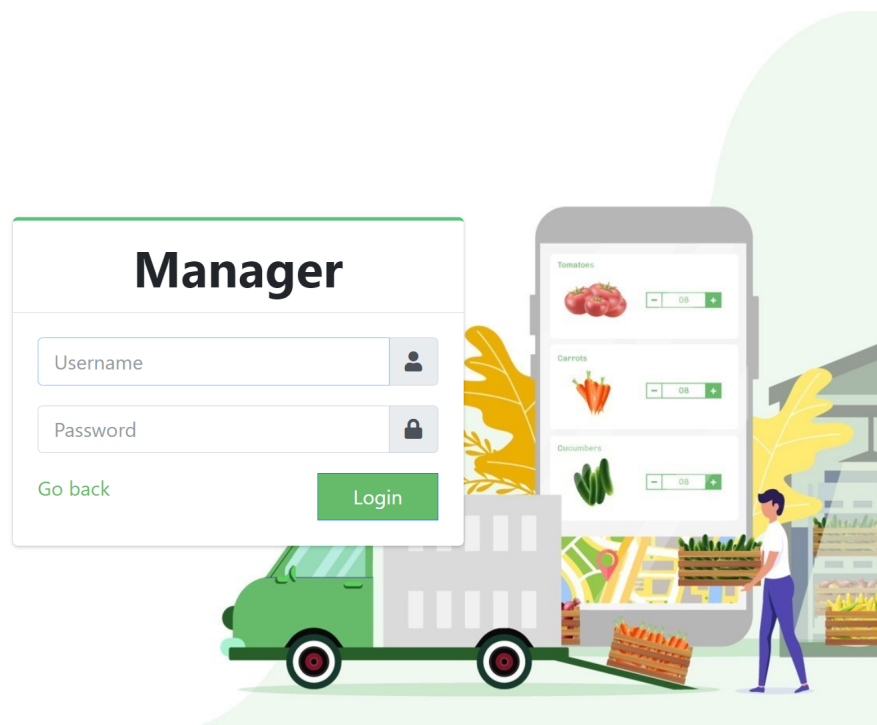
```
?>
```

V. Application screenshot

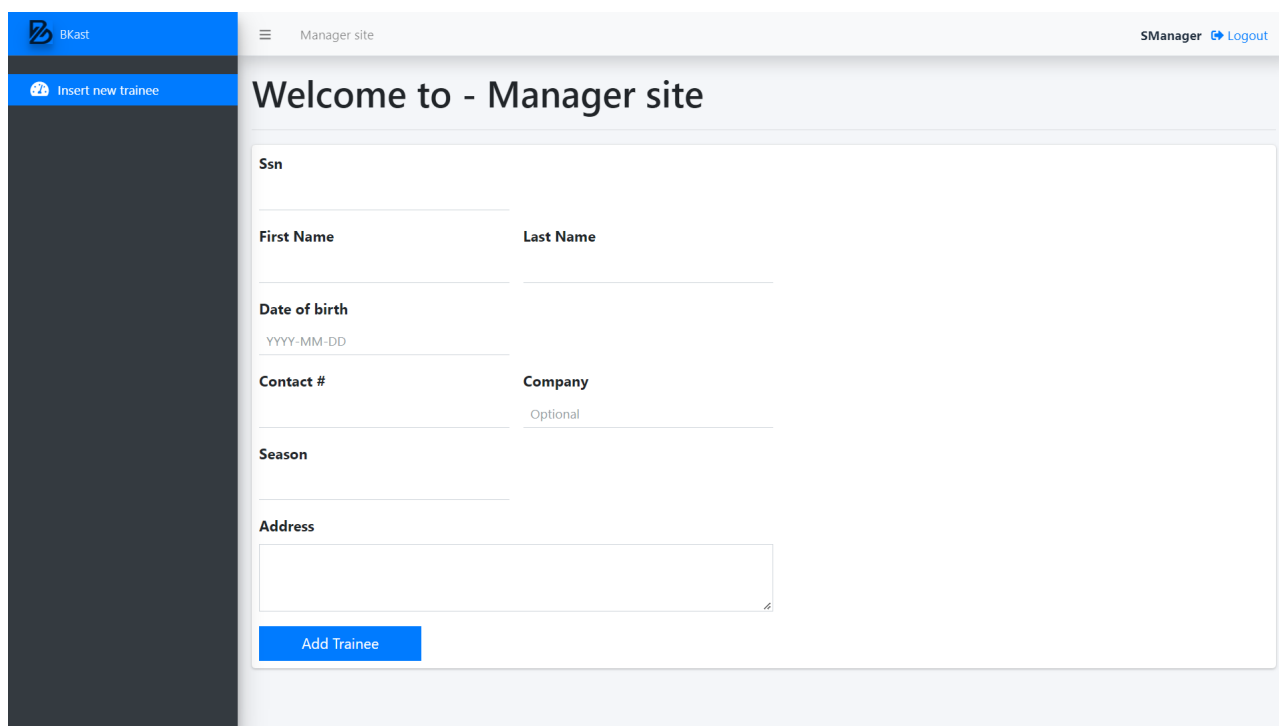
1. Main page



2. Login for Manager

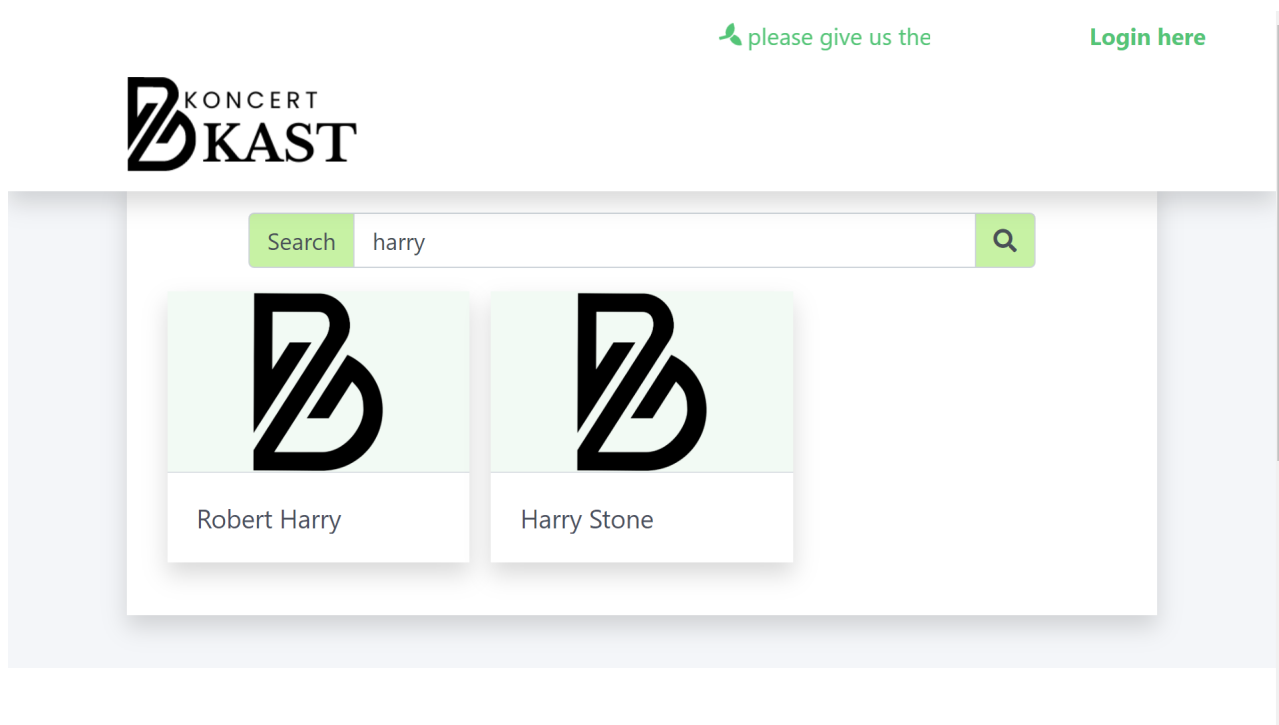


3. Add new information for trainee



The screenshot shows the 'Manager site' interface for BKast. The header includes the BKast logo, a hamburger menu, 'Manager site', and 'SManager Logout'. The left sidebar has a button 'Insert new trainee'. The main content area is titled 'Welcome to - Manager site' and contains a form for adding a new trainee. The form fields are: Ssn, First Name, Last Name, Date of birth (with a YYYY-MM-DD placeholder), Contact #, Company (with an 'Optional' note), Season, and Address. A blue 'Add Trainee' button is at the bottom of the form.

4. The search tool can search for first name, last name, full name or Ssn



The screenshot shows the BKast search results page. At the top, there is a green banner with the text 'please give us the' and a 'Login here' link. Below the banner is the BKast logo. The search bar contains the text 'Search harry' and a magnifying glass icon. Below the search bar, there are two search results displayed as cards. Each card features the BKast logo and a name: 'Robert Harry' and 'Harry Stone'.

please give us the best score you have

[Login here](#)



Search



Harry Stone

please give us

[Login here](#)



Search



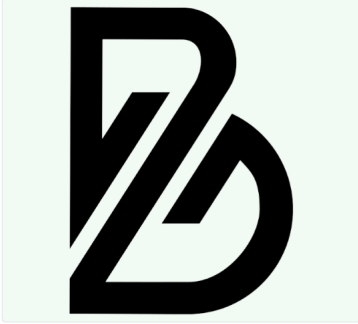
William Charlie


5. The page to show the trainee detail information after select

please give us the best ! [Login here](#)

KONCERT KAST

Trainee's detail information



Luon Vuituoi 

Ssn: 111120180002

Date of birth: 1999-10-23

Company: C101


Phone: 354887694

Address: Cong Quynh Street, District 1, HCMC

Best result season: 2018


Joined season: 1
[2018](#)

6. Clicking the year to retrieve result of a trainee in a season (call a store procedure)

please give us the best score you have  [Login here](#)

KONCERT KAST

Trainee's detail information



Result in season 2018

Episode	Result
1	22.25
2	65
3	470
4	198
5	437

[X Close](#)

Joined season: 1
[2018](#)