

Database Term Project

2022320053

신정윤

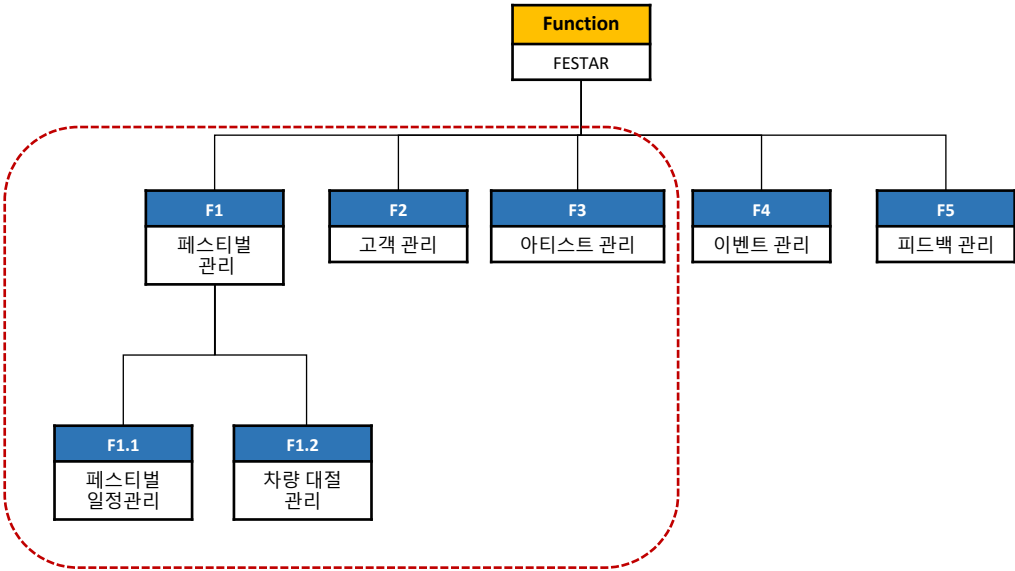
Dept. of Computer Science and Engineering,
College of Informatics, Korea University

Requirement Analysis & Conceptual Design

업무개요서

- ❖ 코로나 19 이후, 서울을 필두로 다양한 공연, 페스티벌이 열리고 있다. 특히, 페스티벌은 하루 종일 다양한 가수가 출연하여 공연한다는 점에서 매력적인 행사로 받아들여진다. 하지만 각 페스티벌의 주최사, 공연자, 홍보 매체 등이 다른 탓에 다양한 정보가 난립하여 보고 싶은 가수가 여러 명인 경우, 어떤 페스티벌에 가야 할지 판단의 어려움이 있다. 이를 근거로 페스티벌을 전체적으로 관리하는 'FESTAR'라는 회사를 운영하고자 한다.
- ❖ 이 회사는 페스티벌에 관한 정보 안내를 총괄적인 업무로 하고, 하위 업무로는 '페스티벌 관리', '고객 관리', "아티스트 관리", '이벤트 관리', '피드백 관리' 등이 필요할 것이다. 이번 프로젝트에서 구현할 업무는 다음과 같다.
- ❖ '페스티벌 관리'는 전반적인 페스티벌에 관한 정보(페스티벌 이름, 날짜, 장소, 대중교통 정보 등)를 관리해야 한다. 이를 '페스티벌 일정 관리' 라고 하며, 우리는 새로운 공연이 확정되면 이를 수집해 데이터를 추가한다. 만약 공연이 취소되는 경우나, 출연진에 변동이 있다면 이 또한 반영해야 한다. 추가적으로 '버스 대절 관리' 라는 업무도 포함되어 있는데, 이는 고객의 거주지역과 멀리서 개최되는 페스티벌에 참여할 수 있도록 소위 '꽃가마'로 불리는 버스 대절 서비스를 제공하기 위한 기능을 한다.
- ❖ '고객 관리'는 FESTAR의 회원 명부에 등록된 사람들을 대상으로 이름, 생년월일, 성별, 나이, 페스티벌 참여 내역, 거주지역 등의 정보를 수집하고 페스티벌에 관한 정보와 연동하여 회원들이 페스티벌에 대한 접근성을 향상시키는 서비스를 제공한다. 또한, 우리 고객들은 페스티벌을 너무 좋아해서, 자신의 거주지와 관련된 페스티벌은 무조건 참여하고, 빠르게 보고 싶어 한다. 이를 반영해야 한다.
- ❖ '아티스트 관리'는 페스티벌에 참여하는 다양한 가수들의 예명, 본명, 본인의 앨범, 본인을 대표하는 3곡 등의 정보를 관리하며, 추가로 과거에 참여한 페스티벌의 이력이나 예정된 페스티벌의 정보와 연동되도록 관리해야 한다.

기능분해도



: 개발 범위

요구사항 명세서

기능명	페스티벌 일정 관리
<div>1. 페스티벌의 정보(페스티벌 이름, 날짜, 장소, 대중교통 정보 등)를 관리한다.</div> <div>2. 페스티벌에 대한 정보가 새롭게 확정되면 상기 정보를 추가한다.<div>a). 정보의 출처를 추가해야 하며, 이는 URL형태로 한 종류만 저장한다. 기본적으로 인스타그램을 이용하며, 없는 경우에는 인터넷 사이트로 한다.</div><div>b). 대중교통 정보는 접근 가능한 다양한 교통수단(기차를 제외한 시내 버스, 지하철 등)을 저장하되, 대중교통이 없는 오지에서 진행되는 경우는 없을 수도 있다.</div></div> <div>3. 페스티벌의 정보는 페스티벌의 이름으로 조회할 수 있다.</div> <div>4. 페스티벌의 정보가 변경되는 경우(취소, 출연진 변동 등), 각각 삭제 및 수정이 가능하다.</div>	

요구사항 명세서

기능명

버스 대절 관리

1. 고객의 거주지와 페스티벌의 개최 장소가 멀리 떨어져 있는 경우, 버스 대절을 통해 페스티벌에 참여하기 쉽도록 한다.
 - a). 이는 페스티벌 일정 관리에서 사용한 대중교통과는 다르며, 여기서는 장거리 이동을 위한 전세버스를 관리하고자 한다.
2. 우리는 버스에 고객이 탑승하는지 여부를 관리할 필요는 없으며, 단순히 고객의 거주지에서 참여하려는 페스티벌 간 왕복 운행하는 버스를 확인하는 기능을 제공하면 된다.
3. 필요한 정보에는 '버스 일련번호', '버스 이름', '출발지', '버스의 수용인원', '버스기사의 이름', '버스기사의 전화번호'가 필요하다.
4. 버스가 부족하다는 가정은 제외하며, 버스의 정보는 '페스티벌 이름'을 통해 조회 할 수 있다.
5. 한대의 버스가 다양한 일자의 페스티벌에 운행될 수 있으며, 같은 버스가 동일한 페스티벌에서 두 번 이상 운행하지는 않는다.
 - a). 즉, 'KU-1호'라는 '버스 이름'의 '버스 일련번호'는 페스티벌마다 다르게 부여된다.
6. 새로운 페스티벌에 배차가 필요한 경우, 새롭게 일련번호를 발급하여 데이터를 삽입한다.
7. 주최측의 사정에 의해서 버스 운행에 변경사항이 생기면 이를 수정, 삭제가 가능하다.

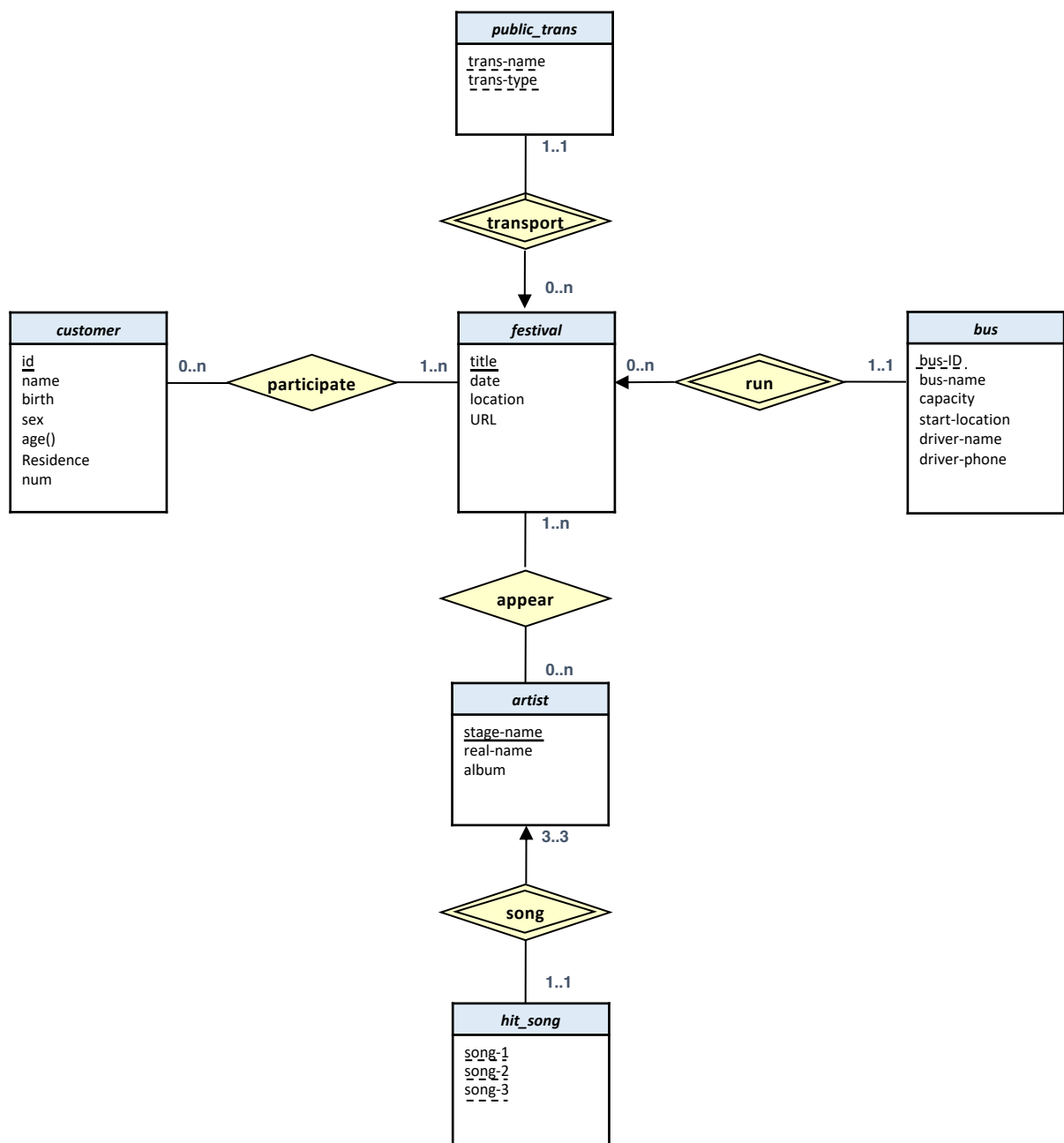
요구사항 명세서

기능명	고객 관리
<div>1. 고객의 정보는 고객의 일련번호, 이름, 생년월일, 성별, 나이, 페스티벌 참여 및 예정 내역, 거주지역 등으로 이루어져 있다.</div> <div>2. 고객이 새로 회원가입을 하게 되면, 위 데이터를 추가해야 한다.</div> <div>3. 고객의 일련번호를 통해서 고객의 정보를 페스티벌 참여 및 예정 내역과 함께 조회한다.</div> <div>4. 고객의 거주지역이 변경될 수 있으므로, 이에 따라 수정이 가능하다.</div> <div>5. 탈퇴 시 고객과 관련된 모든 정보를 삭제해야 한다.</div> <div>6. 고객들의 거주지와 관련된 페스티벌의 참여 내역을 자동으로 추가하고 삭제하는 기능이 필요하다.</div> <div>7. 고객의 페스티벌 참여 횟수를 관리해야 한다.</div>	

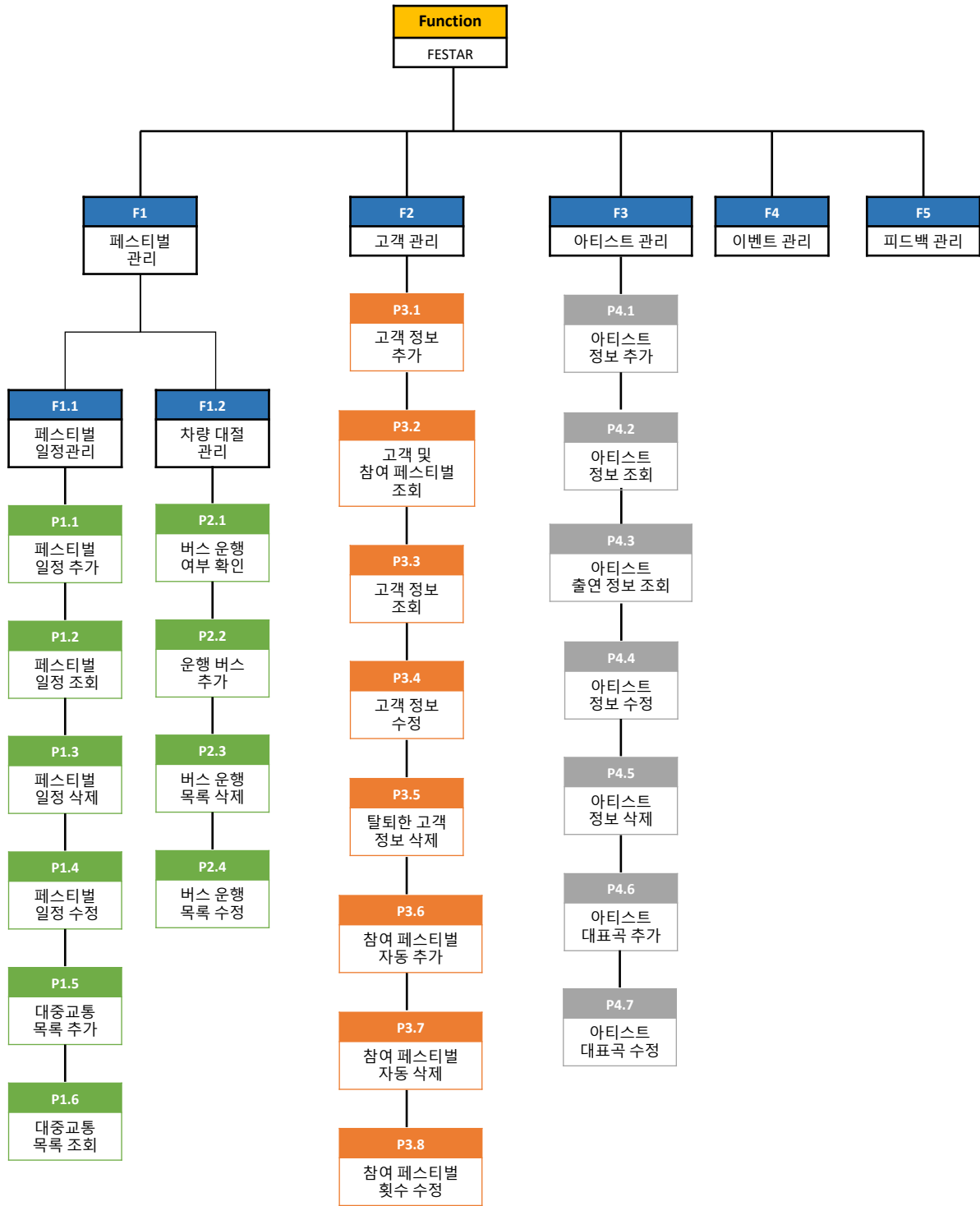
요구사항 명세서

기능명	아티스트 관리
<div>1. 아티스트의 정보는 페스티벌에 참여하는 가수들의 예명, 본명, 본인의 대표 앨범 1장, 본인의 대표곡 3곡으로 구성된다.<div>a). 본명은 공개하지 않을 수 있기에, 데이터가 존재하지 않을 수 있다.</div><div>b). 대표 앨범은 '멜론'의 음원순위가 가장 높은 앨범으로 한다.</div><div>c). 대표곡의 기준은 '멜론'의 음원순위 1,2,3등으로 한다. 이를 실시간으로 관리할 필요는 없고, 3개월 단위로 확인하며 대표곡이 없다는 상황은 제외한다.</div></div> <div>2. 아티스트의 정보를 추가해야 하는 경우는, 상기된 사항을 지키며 추가해야 한다.</div> <div>3. 아티스트의 정보를 조회할 수 있고, 필요하다면 참여 페스티벌의 이름을 조회한다.</div> <div>4. 아티스트 정보가 변경된 경우 데이터를 수정한다.</div> <div>5. 아티스트가 은퇴하거나 10년 이상 활동이 없는 경우 삭제가 가능하다.</div>	

ERD



프로세스 계층도



프로세스 명세서

프로세스명	설명
페스티벌 일정 추가	페스티벌이 확정되면 페스티벌 일정을 추가한다.
페스티벌 일정 조회	페스티벌의 이름을 통해 일정과 관련된 정보를 조회한다.
페스티벌 일정 삭제	페스티벌이 취소되는 경우, 일정 및 대중교통 정보를 삭제한다.
페스티벌 일정 수정	페스티벌의 일정의 변경사항을 수정한다.
대중교통 목록 추가	페스티벌에 연계된 대중교통을 추가한다.
대중교통 목록 조회	페스티벌에 연계된 대중교통을 조회한다.
버스 운행 여부 확인	조회한 페스티벌에서 고객의 거주지와 관련된 버스가 있는지 확인한다.
운행 버스 추가	페스티벌에서 운행할 버스를 추가한다.
버스 운행 목록 삭제	주최측에서 버스를 운행할 필요가 없는 경우, 삭제한다.
버스 운행 목록 수정	버스에 관련된 정보(버스 기사의 정보 등)가 변경되면 수정한다.
고객 정보 추가	회원가입시 회원의 정보를 추가한다.
참여페스티벌 조회	고객의 일련번호를 통해 고객의 참여 페스티벌 내역(과거+예정)을 조회한다.
고객 정보 조회	고객의 일련번호를 통해 고객의 정보를 조회한다.
고객 정보 수정	고객의 개인정보를 수정한다.
탈퇴한 고객 정보 삭제	탈퇴한 고객의 모든 정보를 삭제한다.
참여 페스티벌 자동 추가	거주지와 개최지가 같은 페스티벌은 자동으로 참여한다.
참여 페스티벌 자동 삭제	거주지와 개최지가 다른 페스티벌은 자동으로 삭제하여 빠르게 보여준다.
참여 페스티벌 횟수 수정	고객이 참여한 페스티벌의 횟수를 보여주기 위해, 참여의 변동마다 수정한다.
아티스트 정보 추가	아티스트의 정보(예명, 본명 등)를 추가한다.
아티스트 정보 조회	아티스트의 정보를 조회한다.
아티스트 출연 정보 조회	아티스트의 출연 페스티벌 목록을 조회한다.
아티스트 정보 수정	아티스트의 정보를 수정한다.
아티스트 정보 삭제	아티스트가 은퇴하거나 더 이상 활동을 하지 않는 경우 삭제한다.
아티스트 대표곡 추가	아티스트의 대표곡 3개가 데이터에 없는 경우 추가한다.
아티스트 대표곡 수정	아티스트의 대표곡 3개가 변한 경우(3개월 주기로 확인) 수정한다.

Logical Design

릴레이션 명세서 (1)						
릴레이션명	festival					
릴레이션 설명	페스티벌에 관한 정보					
릴레이션 속성						
속성명	속성정의	타입	길이	제약조건	key	비고
title	페스티벌 이름	char	20	PK	PK	
date	페스티벌 날짜	date		Not Null		
location	페스티벌 위치	varchar	50			
URL	정보 출처	varchar	255	Not Null		대부분의 Instagram 주소는 200자 이상을 초과하지 않음
릴레이션명	public_trans					
릴레이션 설명	페스티벌과 연계된 대중교통 정보					
릴레이션 속성						
속성명	속성정의	타입	길이	제약조건	key	비고
title	페스티벌 이름	char	20	PK	FK	festival (title)
trans-name	대중교통의 이름	varchar	20	PK	PK	
trans-type	대중교통의 종류	varchar	10	PK	PK	

릴레이션 명세서 (2)						
릴레이션명	participate					
릴레이션 설명	고객의 페스티벌 참여 기록					
릴레이션 속성						
속성명	속성정의	타입	길이	제약조건	key	비고
title	페스티벌 이름	char	20	PK	FK	festival (title)
id	고객 일련번호	int		PK	FK	customer (id)
릴레이션명	customer					
릴레이션 설명	고객에 관한 정보					
릴레이션 속성						
속성명	속성정의	타입	길이	제약조건	key	비고
id	고객 일련번호	int		PK	PK	auto_increment
name	고객의 이름	char	10	Not Null		
birth	고객의 생년월일	date		Not Null		
sex	고객의 성별	char	6	Not Null		
age	고객의 나이	int		Not Null		
residence	거주지	varchar	50	Not Null		
Num	페스티벌 참여 횟수	Int				삽입시 0

릴레이션 명세서 (3)

릴레이션명	bus					
릴레이션 설명	버스 대절 관리를 위한 정보					
릴레이션 속성						
속성명	속성정의	타입	길이	제약조건	key	비고
title	페스티벌 이름	char	20	PK	FK	festival (title)
bus-id	버스 일련번호	int		PK	PK	auto_increment
bus-name	버스 이름	varchar	10	Not Null		
capacity	버스 수용인원	int				
start-location	버스 출발지	varchar	50			
driver-name	버스 기사 이름	char	10	Not Null		
driver-phone	버스 기사 전화번호	char	13	Not Null		
릴레이션명	appear					
릴레이션 설명	아티스트의 페스티벌 참여 내역					
릴레이션 속성						
속성명	속성정의	타입	길이	제약조건	key	비고
title	페스티벌 이름	char	20	PK	FK	festival (title)
stage-name	아티스트 예명	varchar	20	PK	FK	artist (stage-name)

릴레이션 명세서 (4)						
릴레이션명	artist					
릴레이션 설명	아티스트에 관한 정보					
릴레이션 속성						
속성명	속성정의	타입	길이	제약조건	key	비고
stage-name	아티스트 예명	varchar	20	PK	PK	
real-name	아티스트 본명	char	10			
album	대표 앨범 이름	varchar	20	Not Null		
릴레이션명	hit-song					
릴레이션 설명	아티스트의 대표 3곡					
릴레이션 속성						
속성명	속성정의	타입	길이	제약조건	key	비고
stage-name	아티스트 예명	varchar	20	PK	FK	artist (stage-name)
song1	1순위 대표곡	varchar	20	PK	PK	
song2	2순위 대표곡	varchar	20	PK	PK	
song3	3순위 대표곡	varchar	20	PK	PK	

모듈 명세서 (1)

모듈명	페스티벌 일정 추가
모듈ID	p.1.1
설명	페스티벌이 확정되면 페스티벌 일정을 추가한다.

관련정보	입력	출력	비고
	페스티벌(festival) -페스티벌 이름 -페스티벌 날짜 -페스티벌 위치 -정보 출처		

모듈 내역

READ 페스티벌: 페스티벌 이름, 페스티벌 날짜, 페스티벌 위치, 정보 출처

INSERT INTO 페스티벌

values 페스티벌 이름, 페스티벌 날짜, 페스티벌 위치, 정보 출처

모듈 명세서 (2)

모듈명	페스티벌 일정 조회		
모듈ID	p.1.2		
설명	페스티벌의 이름을 통해 일정과 관련된 정보를 조회한다.		
관련정보	입력	출력	비고
	페스티벌(festival) -페스티벌 이름	페스티벌(festival) -페스티벌 이름 -페스티벌 날짜 -페스티벌 위치 -정보 출처	

모듈 내역

```
READ 페스티벌: 페스티벌 이름

SELECT *

FROM 페스티벌

WHERE festival-name = "페스티벌 이름"
```

모듈 명세서 (3)

모듈명	페스티벌 일정 삭제		
모듈ID	p.1.3		
설명	페스티벌이 취소되는 경우, 일정 및 대중교통 정보를 삭제한다.		
관련정보	입력	출력	비고
	페스티벌(festival) -페스티벌 이름		

모듈 내역

```
READ 페스티벌: 페스티벌 이름

DELETE FROM 대중교통

WHERE festival-name = "페스티벌 이름"

DELETE FROM 페스티벌

WHERE festival-name = "페스티벌 이름"
```

모듈 명세서 (4)

모듈명	페스티벌 일정 수정		
모듈ID	p.1.4		
설명	페스티벌의 변경사항을 수정한다.		
관련정보	입력	출력	비고
	페스티벌(festival) -페스티벌 이름 -페스티벌 날짜 -페스티벌 위치 -정보 출처		- 수정되는 데이터에 따라 입력이 다를 수 있음.

모듈 내역

READ 페스티벌: 페스티벌 이름, 페스티벌 날짜, 페스티벌 위치, 정보 출처

UPDATE 페스티벌

SET 페스티벌 이름, 페스티벌 날짜, 페스티벌 위치, 정보 출처

모듈 명세서 (5)

모듈명	대중교통 목록 추가
모듈ID	p.1.5
설명	페스티벌에 연계된 대중교통을 추가한다.

관련정보	입력	출력	비고
	페스티벌(festival) -페스티벌 이름 대중교통(public-trans) -대중교통 이름 -대중교통 종류		

모듈 내역

READ 페스티벌: 페스티벌 이름, 대중교통: 대중교통 이름, 대중교통 종류

INSERT INTO 대중교통

values 페스티벌 이름, 대중교통 이름, 대중교통 종류

모듈 명세서 (6)

모듈명	대중교통 목록 조회		
모듈ID	p.1.6		
설명	페스티벌에 연계된 대중교통을 조회한다.		
관련정보	입력	출력	비고
	페스티벌(festival) -페스티벌 이름	페스티벌(festival) -페스티벌 이름 대중교통(public-trans) -대중교통 이름 -대중교통 종류	

모듈 내역

```
READ 페스티벌: 페스티벌 이름

SELECT 페스티벌 이름, 대중교통 이름, 대중교통 종류
FROM 페스티벌 NATURAL JOIN 대중교통
WHERE festival-name = "페스티벌 이름"
```

모듈 명세서 (7)

모듈명	버스 운행 여부 확인		
모듈ID	p.2.1		
설명	조회한 페스티벌에서 고객의 거주지와 관련된 버스가 있는지 확인한다.		
관련정보	입력	출력	비고
	고객(customer) -고객 일련번호 페스티벌(festival) -페스티벌 이름	버스(bus) -버스 이름 -버스 수용인원 -버스 출발지 -버스 기사 이름 -버스 기사 전화번호	

모듈 내역

```
READ 고객: 고객 일련번호, 페스티벌: 페스티벌 이름

SELECT 버스 이름, 버스 수용인원, 버스 출발지, 버스 기사 이름, 버스 기사 전화번호

FROM 페스티벌 NATURAL JOIN 버스

WHERE title = 페스티벌 이름 and 버스 출발지 = (SELECT 거주지 FROM 고객 WHERE id = 고객 일련번호)
```

모듈 명세서 (8)

모듈명	운행 버스 추가
모듈ID	p.2.2
설명	페스티벌에서 운행할 버스를 추가한다.

관련정보	입력	출력	비고
	버스(bus) -버스 일련번호 -버스 이름 -버스 수용인원 -버스 출발지 -버스 기사 이름 -버스 기사 전화번호		- 버스 일련번호는 auto_increment를 통해 생성

모듈 내역

READ 버스: 버스 이름, 버스 수용인원, 버스 출발지, 버스 기사 이름, 버스 기사 전화번호

INSERT INTO 버스

values 버스 일련번호, 버스 이름, 버스 수용인원, 버스 출발지, 버스 기사 이름, 버스 기사 전화번호

모듈 명세서 (9)

모듈명	버스 운행 목록 삭제		
모듈ID	p.2.3		
설명	주최측에서 버스를 운행할 필요가 없는 경우, 삭제한다.		

관련정보	입력	출력	비고
	버스(bus) -버스 일련번호		

모듈 내역

READ 버스: 버스 일련번호

DELETE FROM 버스

WHERE bus-id = 버스 일련번호

모듈 명세서 (10)

모듈명	버스 운행 목록 수정		
모듈ID	p.2.4		
설명	버스에 관련된 정보(버스 기사 정보 등)가 변경되면 수정한다.		
관련정보	입력	출력	비고
	버스(bus) -버스 일련번호 -버스 이름 -버스 수용인원 -버스 출발지 -버스 기사 이름 -버스 기사 전화번호		

모듈 내역

READ 버스: 버스 이름, 버스 수용인원, 버스 출발지, 버스 기사 이름, 버스 기사 전화번호

UPDATE 버스

SET 버스 이름, 버스 수용인원, 버스 출발지, 버스 기사 이름, 버스 기사 전화번호

모듈 명세서 (11)

모듈명	고객 정보 추가		
모듈ID	p.3.1		
설명	회원가입시 회원의 정보를 추가한다.		
관련정보	입력	출력	비고
	고객(customer) -고객 이름 -고객 생년월일 -고객 성별 -고객 나이 -고객 거주지		- 고객 일련번호는 auto_increment를 통해 생성

모듈 내역

READ 고객: 고객 이름, 고객 생년월일, 고객 성별, 고객 나이, 고객 거주지

INSERT INTO 고객

values 고객 일련번호, 고객 이름, 고객 생년월일, 고객 성별, 고객 나이, 고객 거주지

모듈 명세서 (12)			
모듈명	참여 페스티벌 조회		
모듈ID	p.3.2		
설명	고객의 일련번호를 통해 고객의 참여 페스티벌 내역(과거 + 예정)을 조회한다.		
관련정보	입력	출력	비고
	고객(customer) -고객 일련번호	페스티벌(festival) -페스티벌 이름	
모듈 내역			
<div>READ 고객: 고객 일련번호</div> <div>SELECT 페스티벌 이름</div> <div>FROM 참여(participate) NATURAL JOIN 페스티벌</div> <div>WHERE id = 고객 일련번호</div>			

모듈 명세서 (13)

모듈명	고객 정보 조회		
모듈ID	p.3.3		
설명	고객 일련번호를 통해 고객의 정보를 조회한다.		
관련정보	입력	출력	비고
	고객(customer) -고객 일련번호	고객(customer) -고객 일련번호 -고객 이름 -고객 생년월일 -고객 성별 -고객 나이 -고객 거주지	

모듈 내역

```
READ 고객: 고객 일련번호

SELECT 고객 일련번호, 고객 이름, 고객 생년월일, 고객 성별, 고객 나이, 고객 거주지
FROM 고객

WHERE id = 고객 일련번호
```

모듈 명세서 (14)

모듈명	고객 정보 수정		
모듈ID	p.3.4		
설명	고객의 개인정보를 수정한다.		
관련정보	입력	출력	비고
	고객(customer) -고객 일련번호 -고객 이름 -고객 생년월일 -고객 성별 -고객 나이 -고객 거주지		

모듈 내역

READ 고객: 고객 일련번호, 고객 이름, 고객 생년월일, 고객 성별, 고객 나이, 고객 거주지

UPDATE 고객

SET 고객 이름, 고객 생년월일, 고객 성별, 고객 나이, 고객 거주지

모듈 명세서 (15)			
모듈명	탈퇴한 고객 정보 삭제		
모듈ID	p.3.5		
설명	탈퇴한 고객의 모든 정보를 삭제한다.		
관련정보	입력	출력	비고
	고객(customer) -고객 일련번호		
모듈 내역			
READ 고객: 고객 일련번호 DELETE FROM 고객 WHERE id = 고객 일련번호			

모듈 명세서 (16) (추가)

모듈명	참여 페스티벌 자동 추가		
모듈ID	p.3.6		
설명	거주지와 개최지가 같은 페스티벌을 자동으로 참여하게 추가한다.		
관련정보	입력	출력	비고
	고객(customer) -고객 거주지 페스티벌(festival) -개최지		

모듈 내역

READ 고객: 고객 거주지, 페스티벌: 개최지

INSERT INTO 참여

VALUES 고객 일련번호, 페스티벌 이름

페스티벌 LEFT JOIN 고객 on 거주지 == 개최지

모듈 명세서 (17) (추가)

모듈명	참여 페스티벌 자동 삭제		
모듈ID	p.3.7		
설명	거주지와 개최지가 다른 페스티벌은 자동으로 삭제하여 빠르게 출력한다.		
관련정보	입력	출력	비고
	고객(customer) -고객 거주지 페스티벌(festival) -개최지		

모듈 내역

READ 고객: 고객 거주지, 페스티벌: 개최지

DELETE FROM 참여

JOIN 축제 on 페스티벌 이름 == 페스티벌 이름

WHERE 거주지 != 개최지

모듈 명세서 (18) (추가)

모듈명	참여 페스티벌 횟수 수정		
모듈ID	p.3.8		
설명	고객이 참여한 페스티벌의 횟수를 수정한다.		
관련정보	입력	출력	비고
	참여 -페스티벌 이름 -고객 일련번호		

모듈 내역

UPDATE

count from participate

SET num

모듈 명세서 (19)			
모듈명	아티스트 정보 추가		
모듈ID	p.4.1		
설명	아티스트의 정보(예명, 본명 등)를 추가한다.		
관련정보	입력	출력	비고
	아티스트(artist) -아티스트 예명 -아티스트 본명 -대표 앨범 이름		
모듈 내역			
READ 아티스트: 아티스트 예명, 아티스트 본명, 대표 앨범 이름			
INSERT INTO 아티스트			
values 아티스트 예명, 아티스트 본명, 대표 앨범 이름			

모듈 명세서 (20)			
모듈명	아티스트 정보 조회		
모듈ID	p.4.2		
설명	아티스트의 정보를 조회한다		
관련정보	입력	출력	비고
	아티스트(artist) -아티스트 예명	아티스트(artist) -아티스트 예명 -아티스트 본명 -대표 앨범 이름	
모듈 내역			
<div>READ 아티스트: 아티스트 예명</div> <div>SELECT 아티스트 예명, 아티스트 본명, 대표 앨범 이름</div> <div>FROM 아티스트</div> <div>WHERE stage-name = 아티스트 예명</div>			

모듈 명세서 (21)			
모듈명	아티스트 출연 정보 조회		
모듈ID	p.4.3		
설명	아티스트의 출연 페스티벌 목록을 조회한다.		
관련정보	입력	출력	비고
	아티스트(artist) -아티스트 예명	페스티벌(festival) -페스티벌 이름	
모듈 내역			
<pre>READ 아티스트: 아티스트 예명 SELECT 페스티벌 이름 FROM 아티스트 NATURL JOIN 페스티벌 WHERE stage-name = 아티스트 예명 ORDER BY 페스티벌 날짜 desc</pre>			

모듈 명세서 (22)			
모듈명	아티스트 정보 수정		
모듈ID	p.4.4		
설명	아티스트의 정보를 수정한다.		
관련정보	입력	출력	비고
	아티스트(artist) -아티스트 예명 -아티스트 본명 -대표 앨범 이름		
모듈 내역			
READ 아티스트: 아티스트 예명, 아티스트 본명, 대표 앨범 이름			
UPDATE 아티스트			
SET 아티스트 예명, 아티스트 본명, 대표 앨범 이름			

모듈 명세서 (23)			
모듈명	아티스트 정보 삭제		
모듈ID	p.4.5		
설명	아티스트가 은퇴하거나 더 이상 활동을 하지 않는 경우 삭제한다.		
관련정보	입력	출력	비고
	아티스트(artist) -아티스트 예명		
모듈 내역			
READ 아티스트: 아티스트 예명			
DELETE FROM 대표곡			
WHERE stage-name = 아티스트 예명			
DELETE FROM 아티스트			
WHERE stage-name = 아티스트 예명			

모듈 명세서 (24)			
모듈명	아티스트 대표곡 추가		
모듈ID	p.4.6		
설명	아티스트의 대표곡 3개가 데이터에 없는 경우 추가한다.		
관련정보	입력	출력	비고
	아티스트(artist) -아티스트 예명 대표곡(hit-song) -1순위 대표곡 -2순위 대표곡 -3순위 대표곡		
모듈 내역			
READ 아티스트: 아티스트 예명, 대표곡: 1순위 대표곡, 2순위 대표곡, 3순위 대표곡			
INSERT INTO 대표곡			
values 아티스트 예명, 1순위 대표곡, 2순위 대표곡, 3순위 대표곡			

모듈 명세서 (25)			
모듈명	아티스트 대표곡 수정		
모듈ID	p.4.7		
설명	아티스트의 대표곡 3개가 변한 경우(3개월 주기로 확인) 수정한다.		
관련정보	입력	출력	비고
	아티스트(artist) -아티스트 예명 대표곡(hit-song) -1순위 대표곡 -2순위 대표곡 -3순위 대표곡		
모듈 내역			
READ 아티스트: 아티스트 예명, 대표곡: 1순위 대표곡, 2순위 대표곡, 3순위 대표곡			
UPDATE 아티스트			
SET 대표곡: 1순위 대표곡, 2순위 대표곡, 3순위 대표곡			

System Implementation

DB 구현 내역서

테이블명

festival

테이블 스키마

```
MariaDB [db2022320053]> describe festival;
```

Field	Type	Null	Key	Default	Extra
title	char(20)	NO	PRI		
date	date	NO		NULL	
location	varchar(50)	YES		NULL	
URL	varchar(255)	NO		NULL	

4 rows in set (0.00 sec)

테이블 데이터

```
MariaDB [db2022320053]> select * from festival;
```

title	date	location	URL
KT&G 2024 SANGSANG	2024-04-27	Chuncheon	https://www.instagram.com/ssmadang.official/
KUrescendo	2022-05-24	Seoul	https://www.instagram.com/kurescendo_2022/
megafield	2024-06-15	Seoul	https://www.instagram.com/megafield_musicfestival/
pentaport	2024-08-04	Incheon	https://www.instagram.com/pentaportrf/
seouljazz	2024-05-31	Seoul	https://www.instagram.com/seouljazzfestival/
someday	2024-09-07	Seoul	https://www.instagram.com/somedayfestival/

6 rows in set (0.00 sec)

DB 구현 내역서

테이블명 public_trans

테이블 스키마

```
[MariaDB [db2022320053]> describe public_trans;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| title      | char(20)      | NO   | PRI |          |       |
| trans_name | varchar(20)   | NO   | PRI |          |       |
| trans_type | varchar(10)   | NO   | PRI |          |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

테이블 데이터

```
[MariaDB [db2022320053]> select * from public_trans;
+-----+-----+-----+
| title      | trans_name          | trans_type |
+-----+-----+-----+
| KUrescendo | 서울 지하철 1호선   | 지하철    |
| megafield  | 서울 버스 229       | 버스      |
| megafield  | 서울 지하철 2호선   | 지하철    |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

DB 구현 내역서

테이블명

participate

테이블 스키마

```
[MariaDB [db2022320053]> describe participate;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| title | char(20)  | NO   | PRI |          |       |
| id    | int(11)   | NO   | PRI | 0        |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

테이블 데이터

```
[MariaDB [db2022320053]> select * from participate;
+-----+-----+
| title      | id |
+-----+-----+
| KUrescendo | 1  |
| KUrescendo | 6  |
| megafield  | 1  |
| megafield  | 6  |
| pentaport  | 2  |
| pentaport  | 6  |
| seouljazz  | 1  |
| seouljazz  | 6  |
| someday    | 1  |
| someday    | 6  |
+-----+-----+
10 rows in set (0.01 sec)
```

DB 구현 내역서

테이블명

customer

테이블 스키마

```
[MariaDB [db2022320053]> describe customer;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
name	char(10)	NO		NULL	
birth	date	NO		NULL	
sex	char(6)	NO		NULL	
age	int(11)	NO		NULL	
residence	varchar(50)	NO		NULL	
num	int(11)	YES		NULL	

7 rows in set (0.00 sec)

테이블 데이터

```
[MariaDB [db2022320053]> select * from customer;
```

id	name	birth	sex	age	residence	num
1	Lee	2002-02-23	Male	23	Seoul	5
2	Shin	1997-11-21	Male	28	Incheon	1
3	Park	2003-02-12	Male	22	Busan	0
4	Song	1999-06-23	Male	26	Pohang	0
5	Kim	1998-07-14	Male	27	Gwangju	0
6	Roh	2001-05-08	Male	24	Seoul	6

6 rows in set (0.01 sec)

DB 구현 내역서

테이블명

bus

테이블 스키마

```
MariaDB [db2022320053]> describe bus;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| bus_id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| title          | char(20)      | YES  | MUL | NULL    |                 |
| bus_name       | varchar(10)   | NO   |     | NULL    |                 |
| capacity       | int(11)       | YES  |     | NULL    |                 |
| start_location | varchar(50)   | YES  |     | NULL    |                 |
| driver_name    | char(10)      | NO   |     | NULL    |                 |
| driver_phone   | char(13)      | NO   |     | NULL    |                 |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

MariaDB [db2022320053]>
```

테이블 데이터

```
MariaDB [db2022320053]> select * from bus;
+-----+-----+-----+-----+-----+-----+-----+
| bus_id | title          | bus_name | capacity | start_location | driver_name | driver_phone |
+-----+-----+-----+-----+-----+-----+-----+
| 3      | seouljazz     | sj-01    | 3        | Busan         | AAB         | 010-2444-2378 |
| 5      | someday       | sd-01    | 25       | Incheon       | BBB         | 010-2444-2347 |
| 6      | KUrescendo    | KU-01    | 21       | Gwangju       | KIM         | 010-2884-2378 |
| 11     | KT&G 2024 SANGSANG | KT-01    | 24       | Incheon       | ABC         | 010-3478-3897 |
| 14     | seouljazz     | sj-02    | 10       | Gwangju       | AAA         | 010-1111-2222 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

DB 구현 내역서

테이블명 appear

테이블 스키마

```
[MariaDB [db2022320053]> describe appear;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| title      | char(20)      | YES  | MUL | NULL    |       |
| stage_name | varchar(20)   | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

테이블 데이터

```
[MariaDB [db2022320053]> select * from appear;
+-----+-----+
| title      | stage_name |
+-----+-----+
| megafield  | Bol4       |
| KUrescendo | Bol4       |
+-----+-----+
2 rows in set (0.00 sec)
```


DB 구현 내역서

테이블명

artist

테이블 스키마

```
[MariaDB [db2022320053]> describe artist;
```

Field	Type	Null	Key	Default	Extra
stage_name	varchar(20)	NO	PRI	NULL	
real_name	char(10)	YES		NULL	
album	varchar(20)	NO		NULL	

3 rows in set (0.00 sec)

테이블 데이터

```
[MariaDB [db2022320053]> select * from artist;
```

stage_name	real_name	album
BIGBANG		MADE
Bol4	An Jiyeong	RED PLANET
IU	Lee Jieun	Lilac
Yebit	Gong Yebin	23

4 rows in set (0.00 sec)

DB 구현 내역서

테이블명 hit_song

테이블 스키마

```
MariaDB [db2022320053]> describe hit_song;
```

Field	Type	Null	Key	Default	Extra
stage_name	varchar(20)	NO	PRI		
song1	varchar(20)	NO	PRI		
song2	varchar(20)	NO	PRI		
song3	varchar(20)	NO	PRI		

4 rows in set (0.00 sec)

테이블 데이터

```
MariaDB [db2022320053]> select * from hit_song;
```

stage_name	song1	song2	song3
BIGBANG	거짓말	Blue	Bang Bang Bang
Bol4	나의 사춘기에게	여행	우주를줄게
IU	Love Wins all	Blueming	Celebrity
Yebit	On-Air	사랑할거야	누군가의마음이되면

4 rows in set (0.00 sec)

소스코드 요약 설명서

Code	Module	Description
styles.css	N/A	css의 정보를 저장, 다지인적으로 필요한 버튼이나 container의 정보를 추가하였음.
header.php	N/A	k-mall의 형태와 비슷하게 각 관리에 접근할 수 있는 php.
index.php	N/A	사용자가 접근하는 첫번째 화면으로, header.php를 통해 다른 관리에 접근할 수 있게 한다.
festiva_list.php	p.1.2	페스티벌의 일정을 select문을 통해 간략하게 모두 출력한다. 추가로 검색창을 만들어 페스티벌의 이름을 입력하면 일치하는 페스티벌만 보여준다.
festival.php	p.1.2	페스티벌의 '세부사항'을 나타낸다. 또한 이 화면에서 페스티벌에 관련된 여러 관리(버스, 아티스트 참여, 대중교통 정보)를 조회할 수 있다.
festival_insert.php	p.1.1	페스티벌(페스티벌 제목, 날짜, 개최지, URL)을 festival table에 insert하고, 성공적으로 삽입되었다는 메시지를 출력 후, festival_list로 돌아간다.
festival_update.php	p.1.4	페스티벌의 변경점을 업데이트하고, 저장한 후 festival_list로 돌아간다.
festival_delete.php	p.1.3	페스티벌을 삭제한다. 더불어 appear, participate 테이블의 정보와 연결되어 있다면, 함께 삭제한다.
public_insert.php	p.1.5	대중교통(페스티벌 제목, 대중교통 이름, 대중교통 종류)을 public_trans 테이블에 삽입한다.
bus_list.php	N/A	페스티벌마다 운행하는 버스의 전체 목록을 보여준다. 또한, 페스티벌의 제목을 검색하면 그에 맞는 버스의 목록을 보여준다.
bus_insert.php	p.2.2	버스(페스티벌 제목, 버스 이름, 수용량, 출발지, 운전기사 이름, 전화번호)를 bus에 insert한다. 그 후 bus_list.php로 되돌아간다.
bus_update.php	p.2.4	버스의 변경점을 업데이트하고, 저장한 후 bus_list로 돌아간다.
bus_delete.php	p.2.3	버스의 정보를 삭제한다.
search_bus.php	p.2.1	고객의 id로 이 페스티벌에서 고객의 거주지와 출발지가 같은 버스를 조회한다.

소스코드 요약 설명서

Code	Module	Description
customer_list.php	p.3.3	고객의 목록을 select문을 통해 출력하고, 추가 검색 기능을 통해 고객 id를 통해 검색할 수 있다.
customer_insert.php	p.3.1	고객(이름, 생년월일, 성별, 나이, 거주지, 페스티벌 참여 횟수)를 customer에 insert한다.
customer_update.php	p.3.4	고객의 정보를 업데이트한다.
customer_delete.php	p.3.5	고객의 정보를 삭제한다. 이 과정에서 participate에 관련 정보가 있다면 함께 삭제한다.
participate.php	p.3.2	고객이 페스티벌에 참여한 관계를 나타내는 테이블이다.
join.php	p.3.2	고객이 페스티벌에 참여한 관계를 나타내기 위해서, 관련 정보들을 participate 테이블에 삽입할 수 있는 코드이다.
update_location.php	p.3.6	고객의 거주지와 페스티벌의 개최지가 같으면 자동으로 participate에 insert 한다.
delete_location.php	p.3.7	고객의 거주지와 페스티벌의 개최지가 다르면 자동으로 participate에 delete 한다.
count.php	p.3.8	고객의 participate 정보를 통해, 고객의 페스티벌 참여 횟수를 나타낸다.

소스코드 요약 설명서

Code	Module	Description
artist_list.php	p.4.2	아티스트의 목록을 select를 통해 조회하고, 아티스트 예명을 통해 검색하는 기능을 제공한다.
artist_insert.php	p.4.1	아티스트(예명, 본명, 대표 앨범)을 artist에 insert한다.
artist_update.php	p.4.4	아티스트의 정보를 수정한다.
artist_delete.php	p.4.5	아티스트의 정보를 삭제하며, 동시에 히트곡에 대한 정보도 삭제한다.
artist_appear.php	p.4.3	아티스트의 페스티벌 참여 내역을 조회하는 부분이다.
appear.php	N/A	현재 페스티벌에 참여한 아티스트들의 목록을 보여주며, 선택한 아티스트를 목록에 참여할 수 있도록 정보를 appear_insert.php에 전달한다.
hit_song.php	N/A	히트곡을 artist_list의 세부 페이지에서 보여주기 위한 용도, 히트곡의 정보를 간단하게 보여줌
hit_add.php	p.4.6	히트곡을 새롭게 추가하는 코드
hit_update.php	p.4.7	히트곡의 변경사항을 수정하는 코드
appear_insert.php	N/A	appear 테이블에 아티스트의 페스티벌 참여 정보를 insert한다.

프로젝트 변경점

title의 auto_increment	사용할 필요가 없기에 수정함.
p.3.6/7/8	join해서 처리하는 과정을 추가하고, 필요한 정보(참여 횟수 등)를 보여주기 위해 추가함.