Data management system for a music festival

Team 1:
Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

Visitors

- Enter the festival using a ticket
- Have a wristband for shopping/authorization
- Live in the camping area
- Listen to bands in the concert area
- Use the management system to create a timetable of concerts they want to attend
- All operations involving wristband must be stored in the database

Providers

- Providers supply the festival with various content
- Bands:
  - ➢ Songs, merchandise, press information
- Sponsors:
  - ➢ Vendors - pay for selling their products
  - ➢ Donors - pay for placing their advertisements
- Each purchase needs to be stored in the database
- Database needs to allow detailed finance analysis

Organisers

- Choose time and location of the festival
- Make decisions regarding band and sponsor applications
- Alter the ticket prices
- Plan employee shifts
- Database needs to store live statistics for the organisers to make on time decisions

Queries: What is the price a specific visitor paid?

```
Input: Visitor_ID

SELECT ticket_price.price, visitor.visitor_ID, visitor.first_name, visitor.last_name
FROM visitor, ticket, ticket_price
WHERE visitor.visitor_ID = 2
AND visitor.visitor_ID = ticket.visitor_ID
AND ticket.ticket_type_id = ticket_price.ticket_type_id
AND ticket.booking_date >= ticket_price.valid_from
AND ticket.booking_date <= coalesce(ticket_price.valid_to, current_date);
```

Ausgabefeld

Datenanzeige Zerlegung Meldungen Historie

	price numeric(6,2)	visitor_id integer	first_name character varying(100)	last_name character varying(100)
1	50.00	2	Casie	Haygood

OK. Unix Z 7 Sp 74 Bu 373 1 row. 19 ms...

Queries: How many bands got accepted?

```
SELECT
        (SELECT COUNT(*)
        FROM application
        WHERE type='band') AS BandsApplied,
        (SELECT COUNT(*)
        FROM application
        WHERE type='band'
        AND status='ok') AS BandsAccepted;
```

Ausgabefeld

Datenanzeige Zerlegung Meldungen Historie

	bandsapplied bigint	bandsaccepted bigint
1	10	8

Unix Z 1 Sp 1 Bu 1 1 row. 18 ms.

Queries: Which ticket types have been sold the least?

```
WITH sold_tickets AS (
        SELECT COUNT (ticket.ticket_id) AS number,
        ticket_type.type AS type
        FROM ticket, ticket_type
        WHERE ticket.ticket_type_id =
ticket_type.ticket_type_id
        GROUP BY ticket_type.type
)
SELECT sold_tickets.*
FROM sold_tickets
WHERE sold_tickets.number =
  (SELECT MIN(sold_tickets.number)
  FROM sold_tickets);
```

Ausgabefeld

Datenanzeige Zerlegung Meldungen Historie

	number bigint	type character varying(100)
1	1	Sunday Only
2	1	Saturday Only
3	1	VIP All Inclusive

Unix Z 11 Sp 24 Bu 354 3 rows. 20 ms.

Queries: How much money have the bands earned with their merchandise?

```
SELECT prov.name, SUM(s.quantity * prod.price) AS
total
FROM sale s, product prod, provider prov, band b
WHERE s.product_id = prod.product_id
AND prod.provider_id = prov.provider_id
AND prov.provider_id = b.provider_id
GROUP BY prov.name;
```

Ausgabefeld

Datenanzeige Zerlegung Meldungen Historie

	name character varying(100)	total numeric
1	Megadeth	220.00
2	Anthrax	268.00
3	System of a Down	302.00

Unix Z 1 Sp 1 Bu 1 238 chars 3 rows. 13 ms.

Team 1

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

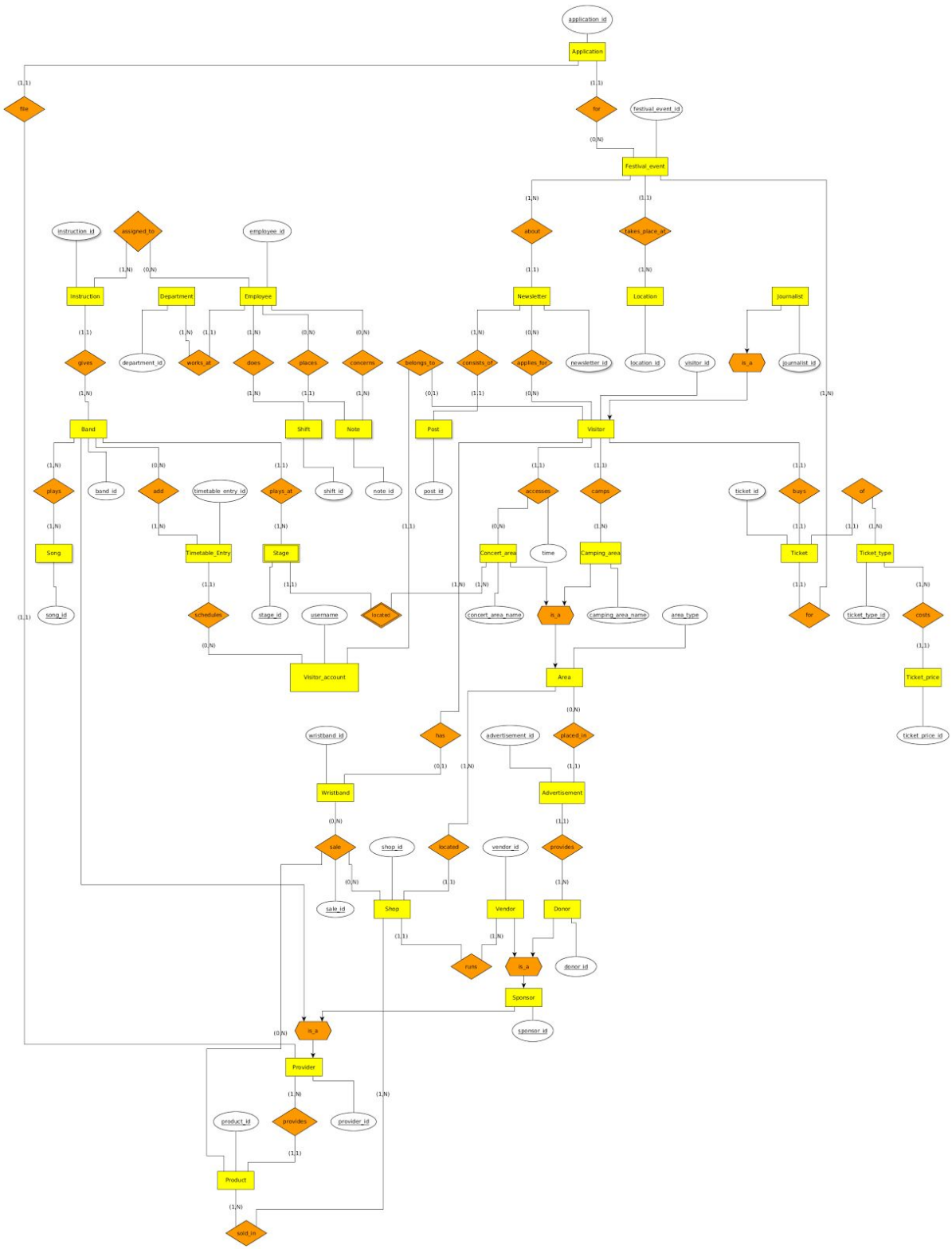
Data Management System for a music festival

1. Concept of presentation

Introduction	Kirill Kldiashvili
ER Model	Christoph Kecht, Fabian Stemmer, Vishesh Mathur
SQL Queries	Thanh Tam Nguyen, Sjaan Arnsfeld

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

2. ER-Model



Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

3. Attributes and Relationships

The following tables contain all attributes which belong to an entity.

Visitor				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
visitor_id	integer	mandatory	identifying	123456
email	string	mandatory	identifying	max.muster@tum.de
last_name	string	mandatory	not	Muster
first_name	string	mandatory	not	Max
address	string	mandatory	not	Ismaniger Str. 22 81675 München
country	string	mandatory	not	Deutschland
birthdate	date	mandatory	not	06.01.1949
phone	string	optional	not	089 1234567
sex	string	optional	not	m

Ticket				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
ticket_id	integer	mandatory	identifying	1234
booking_date	date	mandatory	not	04.05.2017
payment_method	string	mandatory	not	PayPal

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

Wristband				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
wristband_id	integer	mandatory	identifying	0001
disabled	boolean	mandatory	not	Default: true
balance	numeric(6,2)	mandatory	not	Default: 0

Visitor_account				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
username	string	mandatory	identifying	musicfriend24
password	string	mandatory	not	hallo123
filter_date	date	optional	not	It should give a possibility to filter for bands on this specific date.
alarm	timestamp	optional	not	Gives a possibility to set an alarm.

Timetable_Entry				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
timetable_entry_id	integer	mandatory	identifying	12345
preference	integer	mandatory	not	Rating system: Visitor can give his preference for each band from 1(bad) to 5 (best).

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

Band				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
band_id	integer	mandatory	identifying	12345
headliner	boolean	mandatory	not	true
timeslot_date	date	mandatory	not	04.05.2017
timeslot_start	date	mandatory	not	16:00
timeslot_end	date	mandatory	not	17:00
press_information	text	optional	not	Winner of contest XY
is_cancelled	boolean	mandatory	not	If a concert is canceled, the attribute is set to true.

Concert Area				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
concert_area_name	string	mandatory	identifying	Eichenring
concert_capacity	integer	mandatory	not	2000
number_of_visitors	integer	mandatory	not	1546

Stage				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
stage_id	integer	mandatory	identifying	01
name	string	mandatory	not	Zeppelin Stage
capacity	integer	mandatory	not	2000
type	string	mandatory	not	Openair-stage
number_seats	integer	optional	not	0

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

Camping_Area				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
camping_area_name	string	mandatory	identifying	Wolfsresort
camping_capacity	integer	mandatory	not	1000

Provider				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
provider_id	integer	mandatory	identifying	123456
name	string	mandatory	identifying	RedBull

Song				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
song_id	integer	mandatory	identifying	123456
name	string	mandatory	not	Highway To Hell
lyricist	string	mandatory	not	ACDC
length	interval	optional	not	3:12

Instruction				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
instruction_id	integer	mandatory	identifying	123456
text	string	mandatory	not	Turn lead guitar louder
time	timestamp	mandatory	not	2017-05-05 14:37:00

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

Sponsor				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
sponsor_id	integer	mandatory	identifying	123456
money	numeric(9, 2)	mandatory	not	Money given by the sponsor to the festival

Vendor				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
vendor_id	integer	mandatory	identifying	123456

Donor				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
donor_id	integer	mandatory	identifying	123456

Advertisement				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
advertisement_id	integer	mandatory	identifying	123456
type	string	mandatory	not	Poster
quantity	integer	mandatory	not	20

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

Shop				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
shop_id	integer	mandatory	identifying	123456
name	string	mandatory	not	Frank's Würstchenbude
category	string	optional	not	Food

Area				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
area_type	string	mandatory	identifying	"concert" for concert area "camping" for camping area
capacity	integer	mandatory	not	

Product				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
product_id	integer	mandatory	identifying	123456
name	string	mandatory	not	black Shirt
price	numeric(5, 2)	mandatory	not	25.00€
type	string	mandatory	not	T-Shirt or Coke
category	string	mandatory	not	merchandise (or attraction or provision)

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

application				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
application_id	integer	mandatory	identifying	1234567
type	string	mandatory	not	band or vendor or donor
description	string	mandatory	not	Description of the application
date	timestamp	mandatory	not	02-10-2017
status	string	mandatory	not	accepted, rejected, pending

festival_event				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
festival_event_id	integer	mandatory	identifying	1234556
start_date	date	mandatory	not	03-06-2017
end_date	date	mandatory	not	04-06-2017
name	string	mandatory	not	munich music festival

location				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
location_id	integer	mandatory	identifying	1
postcode	string	mandatory	identifying	80993
city	string	mandatory	not	munich
country	string	mandatory	not	germany

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

ticket_type				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
ticket_type_id	integer	mandatory	identifying	1234556
type	string	mandatory	not	VIP/ journalist/ normal
arrival_day	date	mandatory	not	03.05.2017
departure_day	date	mandatory	not	05.05.2017

ticket_price				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
ticket_price_id	integer	mandatory	identifying	1234556
price	numeric(6,2)	mandatory	not	100.00€
valid_from	date	mandatory	not	01.02.2017
valid_to	date	optional	not	31.05.2017

employee				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
employee_id	integer	mandatory	identifying	123455
first_name	string	mandatory	not	Eric
last_name	string	mandatory	not	Schmidt
birthdate	Date	mandatory	not	06.07.1977

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

shift				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
shift_id	integer	mandatory	identifying	Shift in which the employee is working.
type	string	mandatory	not	Example: security, stage area, food counter, etc.
place	string	mandatory	not	near stage
start	timestamp	mandatory	not	2017-05-05 14:00:00
end	timestamp	mandatory	nolt	2017-05-05 22:00:00

department				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
department_id	integer	mandatory	identifying	123456
name	string	mandatory	identifying	Name of the department for which the employee works for.

note				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
note_id	integer	mandatory	identifying	123456
message	string	mandatory	not	
time	timestamp	mandatory	not	The time at which the note was placed.

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

journalist				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
journalist_id	integer	mandatory	identifying	Journalist's id who subscribed for the newsletter.
news_agency	string	mandatory	not	The news agency from which the journalist comes from.

newsletter				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
newsletter_id	string	mandatory	identifying	
title	string	mandatory	not	
published_on	timestamp	mandatory	not	Date at which the newsletter was published.

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

post				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
post_id	integer	mandatory	Identifying	123456
heading	string	mandatory	not	Heading of the newsletter
content	string	mandatory	not	Content of the newsletter
published_time	timestamp	mandatory	not	Time at which the newsletter was published.
tags	string	optional	not	Keywords that the newsletter is about.

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

The following tables contain the attributes of the relationships.

sale				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
sale_id	integer	mandatory	identifying	123456
time	timestamp	mandatory	not	2017-05-05 14:37:00
quantity	integer	mandatory	not	3

accesses				
Attribute name	**Data type**	**mandatory or optional**	**identifying or not**	**comment / example**
time	timestamp	mandatory	not	2017-05-05 14:37:00

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

4. Database schema

```
create table visitor (
        visitor_id      serial primary key,
        email           varchar(100) not null,
        last_name       varchar(100) not null,
        first_name      varchar(100) not null,
        address         varchar(250) not null,
        country         varchar(100) not null,
        birthdate       date not null,
        phone           varchar(50),
        sex             varchar(1)
);


create table journalist (
        journalist_id   serial primary key,
        news_agency varchar(50) not null,
        visitor_id      integer references visitor (visitor_id) not null
);


create table ticket_type (
        ticket_type_id  serial primary key,
        type            varchar(100) not null,
        arrival_day     date not null,
        departure_day date not null
);


create table ticket_price (
        ticket_price_id serial primary key,
        price           numeric(6,2),
        valid_from      date not null,
        valid_to        date,
        ticket_type_id  integer references ticket_type (ticket_type_id) not null
);


create table location (
        location_id     serial primary key,
        postcode        varchar(10) not null,
        city            varchar(100) not null,
        country         varchar(50) not null
);
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

```
create table festival_event (
        festival_event_id       serial primary key,
        start_date              timestamp not null,
        end_date                timestamp not null,
        name                    varchar(50) not null,
        location_id             integer references location (location_id) not null
);


create table application (
        application_id          serial primary key,
        type                    varchar(30) not null,
        description             text not null,
        date                    timestamp not null,
        status                  varchar(15),
        festival_event_id       integer references festival_event (festival_event_id) not null
);


create table provider (
        provider_id       serial primary key,
        name              varchar(100) not null,
        application_id    integer unique references application (application_id) not null
);


create table sponsor (
        sponsor_id      serial primary key,
        provider_id     integer references provider (provider_id) not null,
        money           numeric(9,2) not null
);


create table vendor (
        vendor_id       serial primary key,
        sponsor_id      integer references sponsor (sponsor_id) not null
);


create table donor (
        donor_id        serial primary key,
        sponsor_id      integer references sponsor (sponsor_id) not null
);
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

```
create table area (
        area_type       varchar(100) primary key,
        capacity        integer not null
);


create table advertisement (
        advertisement_id        integer not null,
        type                    varchar(100) not null,
        quantity                integer not null,
        donor_id                integer references donor (donor_id),
        area_type               varchar(100) references area (area_type),
        constraint ad_pk primary key (advertisement_id, area_type)
);


create table stage (
        stage_id        serial primary key,
        name            varchar(100) not null,
        capacity        integer not null,
        type            varchar(200) not null,
        number_seats    integer not null default 0
);


create table band (
        band_id                 serial primary key,
        headliner               boolean not null default false,
        timeslot_date           date not null,
        timeslot_start          time not null,
        timeslot_end            time not null,
        press_information       text,
        is_cancelled            boolean not null default false,
        provider_id             integer references provider (provider_id) not null,
        stage_id                integer references stage (stage_id)
);


create table song (
        song_id                 serial primary key,
        name                    varchar(100) not null,
        lyricist                varchar(100) not null,
        length                  interval
);
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

```
create table visitor_account (
	username	varchar(100) primary key,
	password	varchar(100) not null,
	filter_date	date,
	alarm		timestamp,
	visitor_id	integer unique not null references visitor (visitor_id)
);


create table timetable_entry (
	timetable_entry_id	serial primary key,
	band_id		integer references band (band_id),
	username	varchar(100) references visitor_account (username),
	preference	integer check (preference > 0 and preference < 6)
);


create table product (
	product_id	serial primary key,
	name		varchar(100) not null,
	price		numeric(5,2) not null,
	type		varchar(200) not null,
	category	varchar(200) not null,
	provider_id	integer references provider (provider_id) --null value means that the
product is provided by the festival itself
);


create table shop (
	shop_id		serial primary key,
	name		varchar(100) not null,
	category	varchar(100),
	vendor_id	integer references vendor (vendor_id),
	area_type	varchar(100) references area (area_type)
);


create table sold_in (
	shop_id		integer references shop (shop_id),
	product_id	integer references product (product_id),
	constraint sold_in_pk primary key (shop_id, product_id)
);
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

```
create table wristband (
        wristband_id    serial primary key,
        visitor_id      integer references visitor (visitor_id),
        disabled        boolean not null default true,
        balanc          numeric (6,2) not null default 0.0 check (balance >= 0)
);


create table ticket (
        ticket_id               serial primary key,
        booking_date            date not null,
        payment_method          varchar(50) not null,
        ticket_type_id          integer references ticket_type (ticket_type_id),
        visitor_id              integer references visitor (visitor_id) not null,
        festival_event_id       integer references festival_event (festival_event_id) not null
);


create table department (
        department_id           serial primary key,
        name                    varchar(100) not null
);


create table employee (
        employee_id             serial primary key,
        first_name              varchar(100) not null,
        last_name               varchar(100) not null,
        birthdate               date not null,
        department_id           integer references department (department_id)
);


create table note (
        note_id                 serial primary key,
        message                 text not null,
        time                    timestamp,
        employee_id             integer references employee (employee_id)
);


create table employee_note (
        employee_id     integer references employee (employee_id),
        note_id         integer references note (note_id),
        constraint emp_note_pk primary key (employee_id, note_id)
);
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

```
create table instruction (
        instruction_id   serial primary key,
        text             text not null,
        time             timestamp not null,
        band_id          integer references band (band_id) not null,
        employee_id      integer references employee (employee_id)
);


create table shift (
        shift_id         serial primary key,
        type             varchar(100) not null,
        place            varchar(100) not null,
        start_time       timestamp not null,
        end_time         timestamp not null
);


create table employee_shift (
        employee_id      integer references employee (employee_id),
        shift_id         integer references shift (shift_id),
        constraint emp_shift_pk primary key (employee_id, shift_id)
);


create table newsletter (
        newsletter_id    serial primary key,
        title            varchar(100) not null,
        publish_time     timestamp not null
);


create table newsletter_application (
        visitor_id       integer references visitor (visitor_id),
        newsletter_id    integer references newsletter (newsletter_id),
        constraint visit_news_pk primary key (visitor_id, newsletter_id)
);
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

```
create table post (
        post_id         serial primary key,
        heading         varchar(100) not null,
        content         text not null,
        publish_time    timestamp not null,
        tags            varchar(200),
        newsletter_id   integer references newsletter (newsletter_id)
);


create table sale (
        sale_id         serial primary key,
        time            timestamp not null,
        quantity        integer not null,
        wristband_id    integer references wristband (wristband_id) not null,
        shop_id         integer references shop (shop_id) not null,
        product_id      integer references product (product_id) not null
);


create table area_access (
        wristband_id    integer references wristband (wristband_id) not null,
        area_type       varchar(100) references area (area_type) not null,
        time            timestamp not null,
        constraint wristband_time primary key (wristband_id, time)
);


create table plays (
        song_id         integer not null references song (song_id),
        band_id         integer not null references band (band_id),
        constraint plays_pk primary key (song_id, band_id)
);
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

5. Database indexes

```
CREATE INDEX ON timetable_entry (username);
CREATE INDEX ON timetable_entry (band_id);
CREATE INDEX ON area (area_type);
CREATE INDEX ON product (provider_id);
CREATE INDEX ON vendor (sponsor_id);

CREATE INDEX ON journalist (visitor_id);
CREATE INDEX ON ticket (visitor_id);
CREATE INDEX ON ticket (ticket_type_id);
CREATE INDEX ON ticket_price (ticket_type_id);

CREATE INDEX ON provider (application_id);
CREATE INDEX ON provider (name);
CREATE INDEX ON band (provider_id);
CREATE INDEX ON band (stage_id);
CREATE INDEX ON band (timeslot_date);
CREATE INDEX ON sponsor (provider_id);
CREATE INDEX ON instruction (band_id);
CREATE INDEX ON instruction (employee_id);
CREATE INDEX ON visitor_account (visitor_id);
CREATE INDEX ON visitor (country);
CREATE INDEX ON visitor (last_name);
CREATE INDEX ON plays (song_id);
CREATE INDEX ON plays (band_id);

CREATE INDEX ON donor (sponsor_id);
CREATE INDEX ON advertisement (area_type);
CREATE INDEX ON advertisement (donor_id);
CREATE INDEX ON shop (vendor_id);
CREATE INDEX ON shop (area_type);
CREATE INDEX ON sold_in (product_id);
CREATE INDEX ON wristband (visitor_id);

CREATE INDEX ON sale (wristband_id);
CREATE INDEX ON sale (shop_id);
CREATE INDEX ON sale (product_id);
CREATE INDEX ON post (newsletter_id);
CREATE INDEX ON area_access (wristband_id);
CREATE INDEX ON festival_event (location_id);
CREATE INDEX ON product (name);
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

6. SQL Queries

Visitors:

```
--Price a specific visitor paid for his ticket?
---Input: Visitor_ID

SELECT ticket_price.price, visitor.visitor_ID, visitor.first_name, visitor.last_name
FROM visitor, ticket, ticket_price
WHERE visitor.visitor_ID = 2
AND visitor.visitor_ID = ticket.visitor_ID
and ticket.ticket_type_id = ticket_price.ticket_type_id
and ticket.booking_date >= ticket_price.valid_from
and ticket.booking_date <= coalesce(ticket_price.valid_to, current_date);


--Which ticket-types have been least sold?

WITH sold_tickets AS (
   SELECT COUNT (ticket.ticket_id) AS number, ticket_type.type AS type
   FROM ticket, ticket_type
   WHERE ticket.ticket_type_id = ticket_type.ticket_type_id
   GROUP BY ticket_type.type
)
SELECT sold_tickets.*
FROM sold_tickets
WHERE sold_tickets.number =
   (SELECT MIN(sold_tickets.number)
   FROM sold_tickets);


--Timetable of a specific visitor?
---Input: Visitor_ID

SELECT timetable_entry.*, visitor.first_name, visitor.last_name
FROM visitor, visitor_account, timetable_entry
WHERE visitor.visitor_id = 10
AND visitor.visitor_id = visitor_account.visitor_id
AND visitor_account.username = timetable_entry.username;
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

```
--How many visitors lost their rfid chip?

SELECT COUNT(visitor.*)
FROM visitor, wristband
WHERE wristband.disabled = 'true'
AND wristband.visitor_ID = visitor.visitor_ID;


--How many visitors has the festival?

SELECT COUNT(ticket)
FROM ticket;
```

Providers:

```
--How many shops did each vendor run?

SELECT p.name, COUNT(*)
FROM shop sh, vendor v, sponsor sp, provider p
WHERE sh.vendor_id = v.vendor_id
AND v.sponsor_id = sp.sponsor_id
AND sp.provider_id = p.provider_id
GROUP BY p.name;


--Total time of music played?

SELECT SUM(s.length)
FROM plays ps, song s
WHERE ps.song_id = s.song_id;


--How much money did each visitor spend in total?

SELECT v.last_name, v.first_name, SUM(s.quantity * p.price) + tp.price AS total
FROM sale s, wristband w, visitor v, product p, ticket t, ticket_price tp
WHERE s.wristband_id = w.wristband_id
AND w.visitor_id = v.visitor_id
AND s.product_id = p.product_id
AND v.visitor_id = t.visitor_id
AND t.ticket_type_id = tp.ticket_type_id
AND t.booking_date >= tp.valid_from
AND t.booking_date <= coalesce(tp.valid_to, current_date)
GROUP BY v.last_name, v.first_name, tp.price;
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

```
--How many songs were played in average per band?

WITH total_songs_per_band AS (
   SELECT band_id, COUNT(*) AS songs
   FROM plays
   GROUP BY band_id
)
SELECT AVG(songs) FROM total_songs_per_band;


--How much money did the bands earn with their merchandise?

SELECT prov.name, SUM(s.quantity * prod.price) AS total
FROM sale s, product prod, provider prov, band b
WHERE s.product_id = prod.product_id
AND prod.provider_id = prov.provider_id
AND prov.provider_id = b.provider_id
GROUP BY prov.name;
```

Organisation:

```
--What notes have been placed for a particular department?

SELECT note.message
FROM note, employee, department
WHERE note.employee_id=employee.employee_id
AND employee.department_id=department.department_id
AND department.department_id=1;


--The amount of money collected from a particular type of ticket between certain dates.

SELECT ticket_type.type, SUM(price)
FROM ticket_price, ticket, ticket_type
WHERE ticket_price.ticket_type_id = ticket.ticket_type_id and
ticket.ticket_type_id=ticket_type.ticket_type_id
AND booking_date BETWEEN '2017-05-01' AND '2017-05-31'
AND ticket_price.valid from <= booking_date
AND booking_date <= coalesce (ticket_price.valid_to, current_date)
GROUP BY ticket_type.type;
```

Thanh Tam Nguyen, Sjaan Arnsfeld
Fabian Stemmer, Christoph Kecht
Vishesh Mathur, Kirill Kldiashvili

--On which stage a particular song will be played and between what time?

```
SELECT distinct stage.stage_id, stage.name, band.band_id as BandId, band.timeslot_start
as start, band.timeslot_end as end
FROM band, song, plays, stage
WHERE band.band_id=plays.band_id
AND band.stage_id=stage.stage_id
AND plays.song_id=2;
```

--How many bands got accepted to a particular music festival?

```
SELECT
(SELECT COUNT(*)
FROM application
WHERE type='band') as BandsApplied,
(SELECT COUNT(*)
FROM application
WHERE type='band'
AND status='ok') as BandsAccepted;
```

--Which location has had more than one festival organized?

```
SELECT location_id
FROM festival_event
GROUP BY location_id
HAVING COUNT(location_id)>1;
```