POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EICTA - 1st Project Delivery - ER and SQL

Group 10:

Luigi Tiberio [redacted]
Stefano Moreschi [redacted]
Mattia Silvestri [redacted]

Academic Year: 2025/2026

Contents

Introduction: Provided Text

"A festival is characterized by a name, a location, a start date, and an end date. Each festival is divided into multiple stages, characterized by an identification code, a name, a maximum capacity, and a description about its location within the festival. At each stage, different musical performances can be held. A performance has a unique identifier, a title, a genre (e.g., rock, jazz, electronic), and start/end times. Performances are given by one or more artists, which may be solo musicians or bands (i.e., at least two musicians). The system should store for each artist their stage name, country, and main musical genre. In addition to this, musicians should provide their full name, nationality and main instrument, whereas bands must register their formation year. Alongside music, festivals can host several stands where people can buy food, drinks and merchandise. Each stand has a name, a sponsor, and can sell either of the three aforementioned products. A product is described by a name and a price; furthermore, food and drinks have an expiration date, and only food items have a type (e.g., vegetarian). To enter a festival, people must buy tickets, which have a type (e.g., regular, day pass, VIP), a price, and a validity period. To buy tickets, a person must state their full name, e-mail address, and phone number. A person can buy multiple tickets, and they can also purchase shop items during the festival. Whenever a purchase happens, the system should store the date and the payment method used."

1 | Text Analysis

1.1. Entities

For the creation of the ER model for the Euphoric Events company, we began by identifying the main entities, with their respective attributes, and linking them together through specific relationships. The entities we were able to identify are:

- **Festival**
- **Stage**
- **Performance**
- **Artist**
- **Musician (ISA)**
- **Band (ISA)**
- **Stand**
- **Product**
- **Food (ISA)**
- **Drink (ISA)**
- **Merchandise (ISA)**
- **Person**
- **Purchase**
- **Ticket**

1.2. Attributes

Subsequently, we proceeded to identify the attributes of each entity based on the information provided in the text. Entities are **bolded**; attributes are in *cursive*:

- **Festival**: "A festival is characterized by a *name*, a *location*, a *start date*, and an *end date*."
- **Stage**: "Each festival is divided into multiple stages, characterized by an *identification code*, a *name*, a *maximum capacity*, and a description about its *location* within the festival."
- **Performance**: "A performance has a *unique identifier*, a *title*, a *genre* (e.g., rock, jazz, electronic), and *start/end times*."
- **Artist (with ISA's Musician and Band)**: "The system should store for each artist their *stage name*, *country*, and *main musical genre*. In addition to this, musicians should provide their *full name*, *nationality* and *main instrument*, whereas bands must register their *formation year*."
- **Person**: "To buy tickets, a person must state their *full name*, *e-mail address*, and *phone number*."
- **Ticket**: "To enter a festival, people must buy tickets, which have a *type* (e.g., regular, day pass, VIP), a *price*, and a *validity period*."
- **Purchase**: "Whenever a purchase happens, the system should store the *date* and the *payment method* used."
- **Stand**: "Each stand has a *name*, a *sponsor*, and can sell either of the three aforementioned products."
- **Product (with ISA's Food, Drinks, Merchandise)**: "A product is described by a *name* and a *price*; furthermore, food and drinks have an *expiration date*, and only food items have a *type* (e.g., vegetarian)."

1.3. ISA hierarchies

It is important to note that, during the reasoning process of the different entities, we identified some that are organized hierarchically under parent entities: Musician and Band are linked to the parent Artist, while Food, Drink, and Merchandise derive from Product. In both cases, these are total and exclusive relationships: total because every

Artist must be either a Musician or a Band (there are no "generic" artists); likewise, every Product must be either Food, Drink, or Merchandise (there are no unclassified or hybrid products). They are exclusive because an Artist cannot be both a Band and a Musician at the same time, and a Product cannot simultaneously be Food, Drink, or Merchandise.

1.4. Relationships

Once the entities and their respective attributes were identified, we defined the relationships among the different entities. Each of these connects two entities through a specific relationship characterized by representative cardinalities that indicate how these relationships occur. The relationships (**bolded**) described in the text are:

- **is_divided** (FESTIVAL-STAGE): "Each festival is divided into multiple stages."
- **hold** (STAGE-PERFORMANCE): "At each stage, different musical performances can be held."
- **given_by** (PERFORMANCE-ARTIST): "Performances are given by one or more artists."
- **host** (FESTIVAL-STAND): "Festivals can host several stands."
- **sell** (STAND-PRODUCT): "Each stand can sell either of the three aforementioned products."
- **grant_access_to** (TICKET-FESTIVAL): "To enter a festival, people must buy tickets."
- **make** (PERSON-PURCHASE): "A person can buy..."
- **involve**(PURCHASE-TICKET and PURCHASE-PRODUCT): "...multiple tickets, and they can also purchase shop items during the festival."

1.5. Cardinalities

Important assumptions were made to construct the model from the text; each assumption can be directly deduced from the text provided. The cardinalities of the ER model were built on these assumptions:

- Each Festival must be divided into two or more Stages, while every Stage belongs to exactly one Festival.

- A Stage host different Performances, but each Performance takes place on exactly one Stage.
- A Performance can be given by one or more Artists (Musician or Band), and each Artist can perform in one or more Performances.
- Every Artist record is either Musician or Band, never both. This applies to the data type of the record, not to the real person: the same person may appear both as a solo act and as a member of a band via two distinct Artist records (e.g., "Damiano" as a Musician and "Måneskin" as a Band). Since a Performance may credit multiple Artists, a lineup may list both the band and the solo artist.
- Each Ticket grant access to one specific Festival, while the access to a Festival is granted by one or more Tickets.
- Each Festival can host Stands (i.e.: there can be Festivals without Stands), and every Stand belongs to exactly one Festival.
- A Stand must be able to sell one of the three categories of Product, and a Product can be sold at multiple Stands.
- Every Product must belong to exactly one of the subtypes (Food, Drink, or Merchandise).
- A Person must purchase at least a Ticket to access to the festival, but it is not obligated to purchase Products. A Person is hence obligated to make at least one Purchase (i.e. the Ticket) to access to the Festival and be registered in the database.
- A single Purchase is made by an individual Person.
- A Purchase can involve Tickets (at least one) and Products (if the Person makes a Purchase at a Stand). A single specific Ticket can be bought one time (i.e. involved in one Purchase), while a Product category can be involved in multiple Purchases.

1.6. Introduction of Surrogate Keys:

To construct the model, it was necessary to introduce two new attributes: purchase_id and ticket_id. Both are strings of up to 64 characters and serve as unique identifiers for every Purchase and every Ticket, respectively. The attributes described in the text were not sufficient to form natural primary keys for these entities. Therefore, unique identification codes were introduced to ensure that each record can be unambiguously identified and correctly referenced by foreign keys. Specifically, a Ticket cannot be uniquely identified by

its validity period, type, price, or any combination of these attributes, while a Purchase (i.e., a single transaction) cannot be uniquely identified by its payment method, date, or any combination thereof. For this reason, both entities require surrogate keys to guarantee data integrity and precise referencing. Similarly, it would be possible to introduce a product_id as a surrogate key for Product. However, this addition was deemed unnecessary, since in the current model product_name already functions as a natural key and sufficiently identifies a product type or category (e.g., “Burger”, “Coke”, “T-shirt”). Products are not tracked as individual instances, and the text does not require item-level inventory or product-specific transactions. Therefore, introducing a product_id would only add complexity without providing additional functional benefits for the current scope. Should the system evolve to manage individual product instances, inventories, or stand-specific pricing, the introduction of a product_id could then be justified. The detailed reasoning and justification for all primary key choices are presented in the following sections.

2 | ER Model

2.1. ER Diagram

ER model was developed based on the text analysis presented in the previous section. All entities and relationships were arranged and positioned to capture the most plausible flow of information and interactions as stated in the requirements. The diagram shows graphically how data elements are interrelated in the system, emphasizing hierarchies, cardinalities, and constraints implied in the specification. Cardinalities must be read as entity-cardinality-relationship (e.g.: PERSON (1,N) ->MAKE -> PURCHASE, Person make 1 or more Purchases. PURCHASE (1,1) -> MAKE -> PERSON, a Purchase is made by one Person). Underlined attributes indicate primary keys.

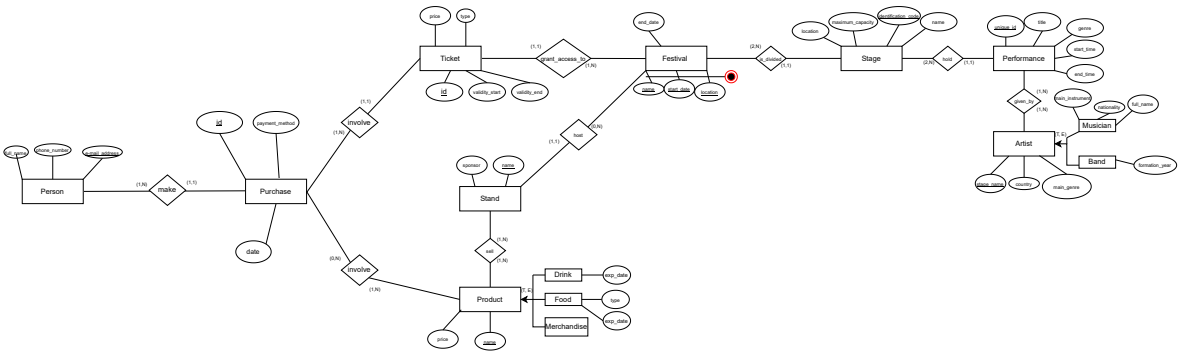

Figure 1. ER diagram for *Euphoric Events*: it models the core entities, attributes, primary keys and their relationships, with representative cardinalities.

2.2. Additional Assumptions, Reasoning and Design Choices

The ER model was introduced to exactly represent the specifications set in the provided text for Euphoric Events. The following design choices have been adopted to have a uniform and significant data structure reflecting the reality of system behaviour. Both the ER and Logical model are thoroughly based on the text analysis reported in chapter 1. We considered modeling Purchase as a relationship between Person and Ticket/Product (i.e., two separate relationships with appropriate cardinalities). However, since the system must record payment method and purchase date for each transaction, Purchase must be modeled as an entity (associative/transaction entity) with its own identifier (purchase_id). This allows one purchase to contain multiple tickets and products, and it cleanly centralizes transaction-level attributes that do not belong to Person, Ticket, or Product. Moreover, as illustrated before, we introduce ticket_id and purchase_id to uniquely distinguish individual tickets and individual transactions, respectively. In this scope, product_name is an adequate natural key, and adding a surrogate key would not provide additional value. An additional possibility is to introduce the many-to-many relationship "compose" between the subtypes Musician and Band. We decided not to include the Band–Musician relationship (band_member or compose table) in our model. Conceptually, this table would represent the membership between Bands and Musicians, storing which musicians belong to which bands (a many-to-many relation) and allowing the system to enforce that each band has at least two musicians, list band line-ups, or prevent double credit if a band and one of its members were both registered in the same performance. However, the original specification does not require the registration of band members: it only asks to distinguish between Musicians and Bands and to store a few additional attributes for each (formation year for Bands; full name, nationality, and main instrument for Musicians). The mention "a band is at least two musicians" serves only to define what a band represents conceptually, not to require recording its individual members. Introducing this relation would therefore add unnecessary complexity, require triggers or maintenance to ensure data integrity, and impose a continuous update effort whenever band compositions change, without adding any real functional value for the system's objectives. The TE specialization between Artist, Musician, and Band already satisfies all the requirements, ensuring that every Artist record is either a Musician or a Band, never both, and that performances can be credited to one or more Artists (e.g. the "Damiano" and "Måneskin" example). For these reasons, the band_member table and its corresponding relation have been intentionally omitted to keep the model simpler, fully

compliant with the text, and focused on the required scope. Additional key assumptions are that Festival editions are uniquely identified by (festival_name, festival_start_date, festival_location) and that Artist stage names are globally unique; otherwise, we would respectively extend the Festival key (or add a surrogate) and use (artist_stage_name, artist_country). These last assumptions will be reprised in the next chapter. Overall, we reckon that the model achieves a faithful and efficient representation of the domain, capturing all functional requirements from the text while avoiding unnecessary complexity or redundancy.

3 | Logical Model

The Logical Model has been built on the same reasoning of the ER model. Attributes where directly extrapolated from the text, except the two created attributes Ticket_id and Purchase_id. The relationships between entities have been thoroughly described in the previous chapters. New bridge tables were built when there was a many-to-many relationship: each of these tables contains the primary keys of the two entities it connects (as foreign keys) and uses their combination as its own primary key, ensuring uniqueness of each pair. For ISA hierarchies, we implemented both the supertype and the corresponding subtype tables. Key choices and additional reasoning are explained under the entity after the arrow. Primary keys are underlined; foreign keys are written in blue:

FESTIVAL (festival_name, festival_start_date, festival_location, festival_end_date)

→ A festival cannot be uniquely identified by name alone: the same name may be used at different locations and times (e.g., "Tomorrowland Brazil" vs "Tomorrowland Belgium"). We identify an edition by (name, start date, location); this allows multiple editions with the same name and location on different start dates (e.g., "Tomorrowland Brazil 19/10–25/10" and "Tomorrowland Brazil 22/10–31/10"). Moreover, it is very unlikely that two editions with the same name and same start date would occur at the same location.

STAGE (festival_name, festival_start_date, festival_location, stage_identification_code, stage_name, stage_location, stage_max_capacity)

→ The text does not guarantee that a stage's identification code is globally unique across all festivals. Therefore, we model stage identity within its parent festival by defining a composite primary key that includes the festival's key.

PERFORMANCE (performance_unique_id, performance_title, performance_genre, performance_start_time, performance_end_time, stage_identification_code, festival_name, festival_start_date, festival_location)

→ In this case, the id is explicitly unique, so it should work as the only primary key.

ARTIST (artist_stage_name, artist_country, artist_main_genre)

→ We reckon that every artist has a different stage name. A possibility is to create a composite primary key with the country.

ARTIST_MUSICIAN (artist_stage_name, musician_full_name, musician_nationality, musician_main_instrument)

ARTIST_BAND (artist_stage_name, artist_stage_name, band_formation_year)

→ The artist subtypes have the same key as the artist supertype and their specific attributes.

PERFORMANCE_ARTIST (performance_unique_id, artist_stage_name)

→ Resolves M:N between Performance and Artist. Indicates which artists do a specific performance, and conversely, in which Performances each Artist performs.

STAND (stand_name, festival_name, festival_start_date, festival_location, stand_sponsor)

→ The text does not guarantee that two different stands at two different festivals cannot have the same stand name. Hence, we have built the composite primary key that includes the specific festival's key.

PRODUCT (product_name, product_price)

→ We consider product name as a sufficient primary key. It indicates a product type or category (e.g.: "Burger", "Coke", "T-shirt") that can be sold at any. If the system wants to register which specific product is purchased or stored, it would be necessary to introduce a unique product_id (analogue to ticket_id).

PRODUCT_DRINK (product_name, drink_exp_date)

PRODUCT_FOOD (product_name, food_exp_date, food_type)

PRODUCT_MERCHANDISE (product_name)

→ The product subtypes have the same key as the product supertype and their specific attributes.

STAND_PRODUCT (stand_name, festival_name, festival_start_date, festival_location, product_name)

→ Resolves M:N between Stand and Product. Indicates which Stand sells which Products, and conversely, which Products are sold at which Stand.

PERSON (person_email_address, person_full_name, person_phone_number)

→ Email addresses work naturally as primary keys, since it uniquely identifies each individual in the system and is already required to purchase tickets or make purchases.

PURCHASE (purchase_id, person_email_address, purchase_date, purchase_payment_method)

→ Purchase must be considered an entity in order to register purchase_date and purchase_payment_method. A unique purchase_id must be created to guarantee a primary key to identify every single transaction that occurs.

TICKET (ticket_id, ticket_type, ticket_price, ticket_validity_start, ticket_validity_end, festival_name, festival_start_date, festival_location, purchase_id)

→ The attributes mentioned in the text (type, price, validity period) are not sufficient to uniquely identify a ticket. Unlike Product, which we model as a category identified by product_name, a Ticket represents a specific instance sold to a buyer, so it needs a surrogate primary key ticket_id. Each ticket must also be associated to exactly one festival, therefore the foreign key (festival_name, festival_start_date, festival_location) is included in Ticket to link the ticket to its festival edition. The relationship to the buyer's purchase is captured via purchase_id (one purchase includes one or more tickets). Alternatively, if tickets were modeled as types (e.g., "Regular", "VIP") rather than instances, we would keep a ticket_type catalog and use a junction table (ticket_purchase) to relate purchases to ticket types.The primary key would then be a complex composite key with Purchase and Festival keys. Having explored both possibilities, we found the ticket_id option easier and more logical and compatible with the text provided.

PRODUCT_PURCHASE (product_name, purchase_id)

→ Resolves M:N between Product and Purchase. Indicates which products are bought by who, and conversely, which people buy which products. As previously discussed, a similar table (TICKET_PURCHASE) could be created following the same logic if tickets were modeled as types (e.g., "Regular", "VIP") instead of instances. In that alternative design, TICKET_PURCHASE would serve the same purpose as PRODUCT_PURCHASE, linking each purchase to the ticket types bought.

4 | SQL

4.1. Coding decision

The database was built in SQL, directly following the structure and constraints defined in the Logical Model. Each entity and relationship in the model is implemented with a CREATE TABLE statement, specifying attributes, primary keys, foreign keys, and any integrity constraints.
Data types used:

- VARCHAR(n): variable-length character strings (100 and 50 used as standard text fields. Alternatively 64 for identifiers, 10 for short coded values, and 254 for email addresses following common practice to ensure sufficient lengths).
- DATE: calendar date (YYYY-MM-DD).
- DATETIME: date + time (YYYY-MM-DD HH:SS used for performance times).
- INT: integer (used for capacities).
- DECIMAL (10,2): fixed-point number for monetary values (price).
- CHECK: simple domain constraints (used only for `purchase_payment_method`, but more possibilities are illustrated).

All columns in this schema are declared NOT NULL, ensuring no incomplete records are stored.

4.2. SQL Tables Definition

```
CREATE TABLE festival (
    festival_name          VARCHAR(100) NOT NULL,
    festival_start_date    DATE NOT NULL,
    festival_location      VARCHAR(100) NOT NULL,
    festival_end_date      DATE NOT NULL,
    PRIMARY KEY (festival_name, festival_start_date, festival_location)
);
```

```
-possible CHECK
(festival_start_date
< festival_end_date)
```

```
CREATE TABLE stage (
    festival_name              VARCHAR(100) NOT NULL,
    festival_start_date        DATE NOT NULL,
    festival_location          VARCHAR(100) NOT NULL,
    stage_identification_code  VARCHAR(100) NOT NULL,
    stage_name                 VARCHAR(100) NOT NULL,
    stage_location             VARCHAR(100) NOT NULL,
    stage_max_capacity         INT NOT NULL,
    PRIMARY KEY (festival_name, festival_start_date, festival_location, stage_identification_code),
    FOREIGN KEY (festival_name, festival_start_date, festival_location)
        REFERENCES festival (festival_name, festival_start_date, festival_location)
);
```

```
CREATE TABLE performance (
    performance_unique_id          VARCHAR(64) NOT NULL,
    performance_title              VARCHAR(100) NOT NULL,
    performance_genre              VARCHAR(100) NOT NULL,
    performance_start_time         DATETIME NOT NULL,
    performance_end_time           DATETIME NOT NULL,
    stage_identification_code      VARCHAR(100) NOT NULL,
    festival_name                  VARCHAR(100) NOT NULL,
    festival_start_date            DATE NOT NULL,
    festival_location              VARCHAR(100) NOT NULL,
    PRIMARY KEY (performance_unique_id),
    FOREIGN KEY (festival_name, festival_start_date, festival_location, stage_identification_cod
        REFERENCES stage (festival_name, festival_start_date, festival_location, stage_identifica
);
```

```
-possible CHECK
(performance_start_time
<
performance_end_time)
```

```
CREATE TABLE artist (
    artist_stage_name          VARCHAR(100) NOT NULL,
    artist_country             VARCHAR(100) NOT NULL,
    artist_main_genre          VARCHAR(100) NOT NULL,
    PRIMARY KEY (artist_stage_name)
);

CREATE TABLE artist_musician (
    artist_stage_name              VARCHAR(100) NOT NULL,
    musician_full_name             VARCHAR(100) NOT NULL,
    musician_nationality           VARCHAR(100) NOT NULL,
    musician_main_instrument       VARCHAR(100) NOT NULL,
    PRIMARY KEY (artist_stage_name),
    FOREIGN KEY (artist_stage_name) REFERENCES artist (artist_stage_name)
);
```

```
CREATE TABLE artist_band (
    artist_stage_name          VARCHAR(100) NOT NULL,
    band_formation_year        INT NOT NULL,
    PRIMARY KEY (artist_stage_name),
    FOREIGN KEY (artist_stage_name) REFERENCES artist (artist_stage_name)
);

CREATE TABLE performance_artist (
    performance_unique_id      VARCHAR(64) NOT NULL,
    artist_stage_name          VARCHAR(100) NOT NULL,
    PRIMARY KEY (performance_unique_id, artist_stage_name),
    FOREIGN KEY (performance_unique_id) REFERENCES performance (performance_unique_id),
    FOREIGN KEY (artist_stage_name) REFERENCES artist (artist_stage_name)
);

CREATE TABLE stand (
    stand_name                 VARCHAR(100) NOT NULL,
    festival_name              VARCHAR(100) NOT NULL,
    festival_start_date        DATE NOT NULL,
    festival_location          VARCHAR(100) NOT NULL,
    stand_sponsor              VARCHAR(100) NOT NULL,
    PRIMARY KEY (stand_name, festival_name, festival_start_date, festival_location),
    FOREIGN KEY (festival_name, festival_start_date, festival_location)
        REFERENCES festival (festival_name, festival_start_date, festival_location)
);

CREATE TABLE product (
    product_name               VARCHAR(100) NOT NULL,
    product_price              DECIMAL(10,2) NOT NULL,
    PRIMARY KEY (product_name)
);

CREATE TABLE product_drink (
    product_name               VARCHAR(100) NOT NULL,
    drink_exp_date             DATE NOT NULL,
    PRIMARY KEY (product_name),
    FOREIGN KEY (product_name) REFERENCES product (product_name)
);
```

```
CREATE TABLE product_food (
    product_name            VARCHAR(100) NOT NULL,
    food_exp_date           DATE NOT NULL,
    food_type               VARCHAR(100) NOT NULL,
    PRIMARY KEY (product_name),
    FOREIGN KEY (product_name) REFERENCES product (product_name)
);

CREATE TABLE product_merchandise (
    product_name            VARCHAR(100) NOT NULL,
    PRIMARY KEY (product_name),
    FOREIGN KEY (product_name) REFERENCES product (product_name)
);

CREATE TABLE stand_product (
    stand_name              VARCHAR(100) NOT NULL,
    festival_name           VARCHAR(100) NOT NULL,
    festival_start_date     DATE NOT NULL,
    festival_location       VARCHAR(100) NOT NULL,
    product_name            VARCHAR(100) NOT NULL,
    PRIMARY KEY (stand_name, festival_name, festival_start_date, festival_location,
product_name),
    FOREIGN KEY (stand_name, festival_name, festival_start_date, festival_location)
        REFERENCES stand (stand_name, festival_name, festival_start_date, festi-
val_location),
    FOREIGN KEY (product_name) REFERENCES product (product_name)
);

CREATE TABLE person (
    person_email_address    VARCHAR(254) NOT NULL,
    person_full_name        VARCHAR(100) NOT NULL,
    person_phone_number     VARCHAR(50) NOT NULL,
    PRIMARY KEY (person_email_address)
);
```

```
CREATE TABLE purchase (
    purchase_id                 VARCHAR(64) NOT NULL,
    person_email_address        VARCHAR(254) NOT NULL,
    purchase_date               DATE NOT NULL,
    purchase_payment_method     VARCHAR(10) NOT NULL
        CHECK (purchase_payment_method IN ('CASH','CARD','ONLINE','OTHER')),
    PRIMARY KEY (purchase_id),
    FOREIGN KEY (person_email_address) REFERENCES person (person_email_address)
);

CREATE TABLE ticket (
    ticket_id               VARCHAR(64) NOT NULL,
    ticket_type             VARCHAR(100) NOT NULL,
    ticket_price            DECIMAL(10,2) NOT NULL,
    ticket_validity_start   DATE NOT NULL,
    ticket_validity_end     DATE NOT NULL,
    festival_name           VARCHAR(100) NOT NULL,
    festival_start_date     DATE NOT NULL,
    festival_location       VARCHAR(100) NOT NULL,
    purchase_id             VARCHAR(64) NOT NULL,
    PRIMARY KEY (ticket_id),
    FOREIGN KEY (festival_name, festival_start_date, festival_location)
        REFERENCES festival (festival_name, festival_start_date, festival_location),
    FOREIGN KEY (purchase_id) REFERENCES purchase (purchase_id)
);
```

```
-possible CHECK
(ticket_type IN
('REGULAR','VIP','DAY-PASS'))
-possible CHECK
(ticket_validity_start <
ticket_validity_end)
```

```
CREATE TABLE product_purchase (
    product_name            VARCHAR(100) NOT NULL,
    purchase_id             VARCHAR(64) NOT NULL,
    PRIMARY KEY (product_name, purchase_id),
    FOREIGN KEY (product_name) REFERENCES product (product_name),
    FOREIGN KEY (purchase_id) REFERENCES purchase (purchase_id)
);
```

5 | Database Population

Mock data were generated mainly using mockaroo and then inserted into the database. Each table contains at least 25 tuples. Given the nature of the domain, some data had to be generated manually (e.g. artist_stage_name). All generated data were manually reviewed and validated for coherence before use. Since some tables contain hundreds of rows of data, only a representative portion of them is pasted in the report. All data screenshots are nevertheless attached to the folder.

festival_name	stival_start_da	estival_locatio	stival_end_date
Sunset Echo	2024-04-21	Tokyo	2024-05-04
Northern Lights	2024-10-09	Madrid	2024-10-18
Velvet Noise	2025-06-18	Barcellona	2025-06-27
Desert Mirage	2024-08-30	Philadelphia	2024-09-11
Rhythm Horizor	2024-02-11	Rome	2024-02-21
Aurora Soundw	2024-03-07	Milan	2024-03-15
Sunset Echo	2025-07-25	Berlin	2025-08-08
Electric Valley	2024-11-12	Barcellona	2024-11-21
Neon Pulse	2025-04-19	Milan	2025-04-26
Velvet Noise	2025-09-04	Tokyo	2025-09-16
Desert Mirage	2024-05-15	Milan	2024-05-28
Cloudstage	2025-12-06	Milan	2025-12-16
Aurora Soundw	2025-09-10	Milan	2025-09-17
Midnight Grove	2024-02-23	Berlin	2024-03-08
Northern Lights	2025-08-13	Monaco	2025-08-22
Sunset Echo	2025-05-28	Naples	2025-06-08
Electric Valley	2024-01-16	Monaco	2024-01-23
Rhythm Horizor	2025-03-02	Rome	2025-03-16
Rhythm Horizor	2024-06-20	Naples	2024-07-02
Neon Pulse	2024-12-27	Monaco	2025-01-05
Velvet Noise	2025-11-09	Naples	2025-11-17
Electric Valley	2024-07-01	Madrid	2024-07-08
Desert Mirage	2025-01-10	Tokyo	2025-01-24
Northern Lights	2024-09-24	Rome	2024-10-05
Midnight Grove	2025-10-03	Rome	2025-10-10

Figure 2. festival table

stival_nam	val_start_	tival_locat	entificatic	tage_nam	age_locatio	_max_capacity
Neon Pulse	2024-12-2	Monaco	STG3646	Beam	Sector 5	80
Sunset Ech	2025-07-2	Berlin	STG9052	Vista	Sector 7	90
Aurora Sos	2024-03-0	Milan	STG6402	Beam	Sector 19	90
Electric Va	2024-07-0	Madrid	STG5619	Luna	Sector 17	60
Neon Pulse	2024-12-2	Monaco	STG7767	Forge	Sector 3	60
Sunset Ech	2025-05-2	Naples	STG2211	Pulse	Sector 16	40
Cloudstage	2025-12-0	Milan	STG2629	Drift	Sector 15	60
Sunset Ech	2024-04-2	Tokyo	STG6670	Peak	Sector 13	40
Northern L	2025-08-1	Monaco	STG3609	Stone	Sector 11	40
Sunset Ech	2025-05-2	Naples	STG5268	Solar	Sector 18	70
Cloudstage	2025-12-0	Milan	STG8845	Stage C	Sector 2	90
Velvet Noi	2025-06-1	Barcellona	STG8817	Bloom	Sector 17	90
Electric Va	2024-11-1	Barcellona	STG9923	Solar	Sector 18	30
Sunset Ech	2025-07-2	Berlin	STG7812	Roots	Sector 4	90
Northern L	2024-10-0	Madrid	STG3827	Peak	Sector 15	100
Rhythm Ha	2024-02-1	Rome	STG3389	Axis	Sector 8	70
Cloudstage	2025-12-0	Milan	STG8949	Vibe	Sector 17	80
Velvet Noi	2025-09-0	Tokyo	STG7642	Beam	Sector 17	70
Neon Pulse	2025-04-1	Milan	STG5310	Forge	Sector 2	90
Northern L	2025-08-1	Monaco	STG6901	Main	Sector 20	70
Midnight C	2024-02-2	Berlin	STG6174	Zen	Sector 11	40
Electric Va	2024-01-1	Monaco	STG1471	Vista	Sector 1	70
Electric Va	2024-11-1	Barcellona	STG7810	Shift	Sector 9	100
Northern L	2024-10-0	Madrid	STG7059	Peak	Sector 18	70
Northern L	2025-08-1	Monaco	STG7619	Arc	Sector 19	60
Rhythm Ha	2024-06-2	Naples	STG9220	Edge	Sector 11	60
Electric Va	2024-07-0	Madrid	STG8997	Nova	Sector 12	30
Midnight C	2024-02-2	Berlin	STG4116	Edge	Sector 20	40
Electric Va	2024-01-1	Monaco	STG6022	Peak	Sector 5	40
Northern L	2024-09-2	Rome	STG5987	Stone	Sector 11	30
Rhythm Ha	2024-02-1	Rome	STG4501	Core	Sector 7	60
Desert Mir	2024-05-1	Milan	STG3021	Beam	Sector 14	70
Desert Mir	2024-05-1	Milan	STG3941	Beam	Sector 3	40
Velvet Noi	2025-09-0	Tokyo	STG5255	Harbor	Sector 6	50
Desert Mir	2025-01-1	Tokyo	STG7456	Core	Sector 9	50
Electric Va	2024-11-1	Barcellona	STG2402	Echo	Sector 10	40
Aurora Sos	2024-03-0	Milan	STG8100	Nexus	Sector 20	80
Aurora Sos	2024-03-0	Milan	STG3521	Stone	Sector 8	50
Rhythm Ha	2024-02-1	Rome	STG9755	Field	Sector 18	60
Northern L	2025-08-1	Monaco	STG5366	Forge	Sector 3	100
Desert Mir	2024-08-3	Philadelph	STG4575	Forge	Sector 13	40
Rhythm Ha	2024-06-2	Naples	STG6283	Bloom	Sector 15	90
Rhythm Ha	2024-02-1	Rome	STG3773	Arc	Sector 17	90
Sunset Ech	2025-07-2	Berlin	STG7386	Axis	Sector 6	80
Rhythm Ha	2025-03-0	Rome	STG2860	Edge	Sector 2	60
Rhythm Ha	2025-03-0	Rome	STG5704	Main	Sector 10	90

Electric Va	2024-11-1	Barcellona	STG3937	Stage A	Sector 20	100
Velvet Noi	2025-09-0	Tokyo	STG1429	Peak	Sector 19	30
Neon Pulse	2025-04-1	Milan	STG6685	Vibe	Sector 9	70
Midnight C	2024-02-2	Berlin	STG2562	Main	Sector 19	60
Rhythm Ha	2025-03-0	Rome	STG2666	Pulse	Sector 8	30
Desert Mir	2025-01-1	Tokyo	STG3038	Echo	Sector 3	100
Aurora Sos	2024-03-0	Milan	STG5086	Drift	Sector 5	100
Velvet Noi	2025-11-0	Naples	STG9568	Edge	Sector 3	30
Northern L	2025-08-1	Monaco	STG4009	Main	Sector 17	70
Aurora Sos	2025-09-1	Milan	STG2679	Rift	Sector 1	80
Electric Va	2024-11-1	Barcellona	STG7057	Echo	Sector 3	60
Sunset Ech	2025-05-2	Naples	STG6862	Shift	Sector 1	70
Midnight C	2024-02-2	Berlin	STG1564	Bloom	Sector 7	60
Midnight C	2025-10-0	Rome	STG4327	Stage A	Sector 1	60
Neon Pulse	2025-04-1	Milan	STG9783	Beam	Sector 9	80
Desert Mir	2025-01-1	Tokyo	STG2121	Rift	Sector 19	40
Velvet Noi	2025-06-1	Barcellona	STG4727	Luna	Sector 13	40
Neon Pulse	2025-04-1	Milan	STG3677	Arc	Sector 11	40
Velvet Noi	2025-11-0	Naples	STG1505	Crest	Sector 20	80
Desert Mir	2024-05-1	Milan	STG7245	Rift	Sector 7	30
Northern L	2024-10-0	Madrid	STG8228	Nexus	Sector 7	40
Northern L	2024-09-2	Rome	STG9424	Harbor	Sector 20	50
Desert Mir	2024-05-1	Milan	STG6061	Forge	Sector 18	30
Velvet Noi	2025-06-1	Barcellona	STG9585	Main	Sector 15	30
Northern L	2024-09-2	Rome	STG4817	Crest	Sector 7	50
Velvet Noi	2025-09-0	Tokyo	STG7431	Roots	Sector 3	60
Sunset Ech	2024-04-2	Tokyo	STG5315	Yard	Sector 19	90
Northern L	2024-09-2	Rome	STG4838	Drift	Sector 15	90
Rhythm Ha	2024-02-1	Rome	STG3459	Echo	Sector 14	70
Desert Mir	2024-08-3	Philadelph	STG2100	Nexus	Sector 6	40
Midnight C	2025-10-0	Rome	STG9511	Zen	Sector 18	100
Aurora Sos	2024-03-0	Milan	STG3833	Core	Sector 11	90
Electric Va	2024-01-1	Monaco	STG3456	Forge	Sector 5	40
Rhythm Ha	2025-03-0	Rome	STG3113	Bloom	Sector 8	40
Rhythm Ha	2024-02-1	Rome	STG3729	Beam	Sector 3	40
Velvet Noi	2025-11-0	Naples	STG4722	Axis	Sector 14	40
Neon Pulse	2024-12-2	Monaco	STG7314	Haze	Sector 11	30
Sunset Ech	2024-04-2	Tokyo	STG7307	Stage A	Sector 2	50
Desert Mir	2024-05-1	Milan	STG3993	Field	Sector 18	80
Aurora Sos	2025-09-1	Milan	STG2522	Roots	Sector 8	40
Midnight C	2025-10-0	Rome	STG3548	Echo	Sector 17	80
Electric Va	2024-01-1	Monaco	STG5865	Drift	Sector 16	80
Velvet Noi	2025-11-0	Naples	STG7162	Haze	Sector 3	80
Midnight C	2025-10-0	Rome	STG3373	Vibe	Sector 15	100
Aurora Sos	2025-09-1	Milan	STG4441	Haze	Sector 6	100
Northern L	2024-10-0	Madrid	STG2734	Drift	Sector 6	90
Velvet Noi	2025-06-1	Barcellona	STG4158	Yard	Sector 6	90

Aurora Sou	2024-03-0	Milan	STG6394	Nexus	Sector 4	50
Neon Pulse	2025-04-1	Milan	STG5590	Axis	Sector 12	90
Midnight C	2024-02-2	Berlin	STG5073	Main	Sector 2	50
Northern L	2024-10-0	Madrid	STG5519	Main	Sector 5	30
Velvet Noi	2025-09-0	Tokyo	STG5423	Forge	Sector 4	70
Northern L	2024-09-2	Rome	STG8589	Pulse	Sector 6	80
Rhythm Hc	2024-02-1	Rome	STG7256	Stage A	Sector 19	100

Figure 3. stage table

mance_uni	formance_	ormance_g	nance_sta	mance_en	dentificatio	stival_nan	ival_start_	tival_location
PFM237	KitKats – E	Electronic	2025-01-0	2025-01-0	STG3646	Neon Pulse	2024-12-2	Monaco
PFM926	Puh R&B S	R&B	2024-12-3	2024-12-3	STG3646	Neon Pulse	2024-12-2	Monaco
PFM557	The Wiene	Country	2025-08-0	2025-08-0	STG9052	Sunset Ech	2025-07-2	Berlin
PFM412	BimBamBu	Metal	2025-08-0	2025-08-0	STG9052	Sunset Ech	2025-07-2	Berlin
PFM956	Niños del r	R&B	2024-03-1	2024-03-1	STG6402	Aurora Sou	2024-03-0	Milan
PFM534	Obrigados	Reggae	2024-03-1	2024-03-1	STG6402	Aurora Sou	2024-03-0	Milan
PFM130	MunichMc	Country	2024-07-0	2024-07-0	STG5619	Electric Va	2024-07-0	Madrid
PFM885	Niños del r	R&B	2024-07-0	2024-07-0	STG5619	Electric Va	2024-07-0	Madrid
PFM737	Μπάστα –	Hip-Hop	2024-12-2	2024-12-2	STG7767	Neon Pulse	2024-12-2	Monaco
PFM973	Obrigados	Reggae	2024-12-2	2024-12-2	STG7767	Neon Pulse	2024-12-2	Monaco
PFM565	Los Enanit	Indie	2025-06-0	2025-06-0	STG2211	Sunset Ech	2025-05-2	Naples
PFM198	Puh R&B S	R&B	2025-06-0	2025-06-0	STG2211	Sunset Ech	2025-05-2	Naples
PFM515	The Freaks	House	2025-12-1	2025-12-1	STG2629	Cloudstage	2025-12-0	Milan
PFM652	The Cockr	Pop	2025-12-1	2025-12-1	STG2629	Cloudstage	2025-12-0	Milan
PFM668	Obrigados	Reggae	2024-04-2	2024-04-2	STG6670	Sunset Ech	2024-04-2	Tokyo
PFM860	Puh R&B S	R&B	2024-05-0	2024-05-0	STG6670	Sunset Ech	2024-04-2	Tokyo
PFM331	Redday Liv	Reggae	2025-08-1	2025-08-1	STG3609	Northern L	2025-08-1	Monaco
PFM181	La revoluçi	Indie	2025-08-1	2025-08-1	STG3609	Northern L	2025-08-1	Monaco
PFM158	La revoluçi	Indie	2025-05-2	2025-05-2	STG5268	Sunset Ech	2025-05-2	Naples
PFM775	The Cockr	Pop	2025-06-0	2025-06-0	STG5268	Sunset Ech	2025-05-2	Naples
PFM663	The Freaks	House	2025-12-1	2025-12-1	STG8845	Cloudstage	2025-12-0	Milan
PFM675	The Wiene	Country	2025-12-1	2025-12-1	STG8845	Cloudstage	2025-12-0	Milan
PFM269	Redday Re	Reggae	2025-06-2	2025-06-2	STG8817	Velvet Noi	2025-06-1	Barcellona
PFM651	Green Colc	R&B	2025-06-2	2025-06-2	STG8817	Velvet Noi	2025-06-1	Barcellona
PFM602	BimBamBu	Metal	2024-11-1	2024-11-1	STG9923	Electric Va	2024-11-1	Barcellona
PFM129	Foo Allies -	Country	2024-11-1	2024-11-1	STG9923	Electric Va	2024-11-1	Barcellona
PFM461	Les Baguet	Folk	2025-07-2	2025-07-2	STG7812	Sunset Ech	2025-07-2	Berlin
PFM993	Μπάστα Η	Hip-Hop	2025-07-2	2025-07-2	STG7812	Sunset Ech	2025-07-2	Berlin
PFM573	Green Colc	R&B	2024-10-1	2024-10-1	STG3827	Northern L	2024-10-0	Madrid
PFM584	Spice Gals	Techno	2024-10-1	2024-10-1	STG3827	Northern L	2024-10-0	Madrid
PFM918	Green Colc	R&B	2024-02-1	2024-02-1	STG3389	Rhythm Hc	2024-02-1	Rome
PFM169	Obrigados	Reggae	2024-02-1	2024-02-1	STG3389	Rhythm Hc	2024-02-1	Rome
PFM137	BimBamBu	Metal	2025-12-0	2025-12-0	STG8949	Cloudstage	2025-12-0	Milan
PFM812	Blancodolc	Electronic	2025-12-1	2025-12-1	STG8949	Cloudstage	2025-12-0	Milan
PFM510	Blancodolc	Electronic	2025-09-0	2025-09-0	STG7642	Velvet Noi	2025-09-0	Tokyo
PFM304	Redday Re	Reggae	2025-09-1	2025-09-1	STG7642	Velvet Noi	2025-09-0	Tokyo
PFM960	The Freaks	House	2025-04-2	2025-04-2	STG5310	Neon Pulse	2025-04-1	Milan
PFM694	The Wiene	Country	2025-04-2	2025-04-2	STG5310	Neon Pulse	2025-04-1	Milan
PFM305	Eiffelboys	Indie	2025-08-2	2025-08-2	STG6901	Northern L	2025-08-1	Monaco
PFM880	The Freaks	House	2025-08-1	2025-08-1	STG6901	Northern L	2025-08-1	Monaco
PFM441	Μπάστα –	Hip-Hop	2024-03-0	2024-03-0	STG6174	Midnight C	2024-02-2	Berlin
PFM296	Linkin park	Hip-Hop	2024-03-0	2024-03-0	STG6174	Midnight C	2024-02-2	Berlin
PFM411	Niños del r	R&B	2024-01-2	2024-01-2	STG1471	Electric Va	2024-01-1	Monaco
PFM281	Blancodolc	Electronic	2024-01-1	2024-01-1	STG1471	Electric Va	2024-01-1	Monaco
PFM709	La revoluçi	Indie	2024-11-1	2024-11-1	STG7810	Electric Va	2024-11-1	Barcellona
PFM391	Galifianaki	Jazz	2024-11-1	2024-11-1	STG7810	Electric Va	2024-11-1	Barcellona

Figure 4. performance table (portion)

artist_stage_name	artist_country	artist_main_genre
Calcattu	Italy	Pop
Tiziano Cobalto	Italy	Rock
Vecchiotti	Italy	Jazz
Vasco Verdi	Italy	Pop
Taylor Quick	USA	Hip Hop
Katy Cherry	USA	Classical
P-Griddy	USA	Pop
Kanye East	USA	Rock
Michael Sackson	USA	Pop
Duplo	Germany	Jazz
Frankenstein	Germany	Hip Hop
Pi-Raf	France	Rock
Stromale	France	Pop
Goofy!	Netherlands	Classical
New Amsterdam	Netherlands	Pop
Ojosdelmar	Spain	Jazz
Jamon Blanco	Spain	Rock
DJ-Guapo	Portugal	Pop
Franco	Brazil	Hip Hop
Taqquinho	Brazil	Pop
$uperman	Brazil	Rock
Miss Messi	Argentina	Jazz
Eurobaby	Belgium	Pop
Good Bunny	Bolivia	Classical
Loredana Bercaffè	San Marino	Pop
I Gatti	Italy	Rock
Puh	Italy	Hip Hop
Blancodolce	Italy	Pop
Eins Zwei Polizei	Germany	Jazz
MunichMonsters	Germany	Rock
The Wienerwursts	Germany	Pop
KitKats	Germany	Classical
Green Cold Soup Beans	USA	Pop
Foo Allies	USA	Rock
The Cockroaches	United Kingdom	Pop
The Freaks	USA	Jazz
Redday	USA	Hip Hop
Linkin parking lot	USA	Pop
Spice Gals	USA	Rock
Obrigados	Portugal	Classical
Disculpe!	Spain	Pop
Eiffelboys	France	Jazz
Les Baguettes	France	Pop
BimBamBum	Brazil	Rock
La revolução	Brazil	Hip Hop
Zeb&friends	Malta	Pop
Μπάστα	Greece	Rock
Galifianakis	Greece	Classical
Niños del mar	Uruguay	Pop
Los Enanitos	Mexico	Jazz

Figure 5. artist table

artist_stage_name	musician_full_nam	musician_nationalitician_main_instrument	
Calcattu	Gaetano Napoli	Italian	Acoustic guitar
Tiziano Cobalto	Tiziano Cobalto	Italian	Flute
Vecchiotti	Roberto Vecchiotti	Italian	Didgeridoo
Vasco Verdi	Mario Rossi	Italian	Drums
Taylor Quick	Taylor Smith	American	Electric guitar
Katy Cherry	Katy Roberts	American	Kazoo
P-Griddy	Tyler White	American	Violin
Kanye East	Kanye West	American	Bass
Michael Sackson	Michael Sackson	American	Classical guitar
Duplo	Sonia Hummels	German	Drums
Frankenstein	Mario Kimmich	German	Flute
Pi-Raf	Pierre Rafael	French	Acoustic guitar
Stromale	Mathieu Macron	French	Electric guitar
Goofy!	Isaac Van Persie	Dutch	Electric guitar
New Amsterdam	Arjen Robben	Dutch	Violin
Ojosdelmar	Daniela de la Cruz	Spanish	Classical guitar
Jamon Blanco	Juan Blanco	Spanish	Classical guitar
DJ-Guapo	Roberto Mal	Portuguese	Classical guitar
Franco	Francisco Bolsa	Brazilian	Acoustic guitar
Taqquinho	Ana Blanco	Brazilian	Electric guitar
Ṣuperman	Mariano Samir San	Brazilian	Violin
Miss Messi	Gabriela Messi	Argentinian	Acoustic guitar
Eurobaby	Lucas Vermeulen	Belgian	Drums
Good Bunny	Juan Trinidad	Bolivian	Didgeridoo
Loredana Bercaffè	Loredana Buonano	Sammarinese	Bass

Figure 6. artist_musician table

artist_stage_name	band_formation_year
I Gatti	1993
Puh	2005
Blancodolce	1998
Eins Zwei Polizei	2011
MunichMonsters	1990
The Wienerwursts	2014
KitKats	2001
Green Cold Soup Bean	1997
Foo Allies	2008
The Cockroaches	1990
The Freaks	2016
Redday	2003
Linkin parking lot	2019
Spice Gals	1995
Obrigados	2011
Disculpe!	2006
Eiffelboys	2007
Les Baguettes	1992
BimBamBum	2005
La revolução	2018
Zeb&friends	1999
Μπάστα	2005
Galifianakis	2013
Niños del mar	2001
Los Enanitos	1994

Figure 7. artist_band table

performance_unique_id	artist_stage_name
PFM189	Foo Allies
PFM189	Los Enanitos
PFM110	The Cockroaches
PFM980	Los Enanitos
PFM679	La revolução
PFM679	Les Baguettes
PFM558	Eiffelboys
PFM742	Eins Zwei Polizei
PFM742	Les Baguettes
PFM719	I Gatti
PFM564	Green Cold Soup Beans
PFM155	Galifianakis
PFM412	BimBamBum
PFM237	KitKats
PFM553	Linkin parking lot
PFM553	Puh
PFM401	I Gatti
PFM401	Obrigados
PFM937	Eiffelboys
PFM948	The Freaks
PFM447	Foo Allies
PFM447	Linkin parking lot
PFM659	Foo Allies
PFM659	Eins Zwei Polizei
PFM450	MunichMonsters
PFM450	Foo Allies
PFM450	Eins Zwei Polizei
PFM335	I Gatti
PFM335	Redday
PFM923	Obrigados
PFM923	Eiffelboys
PFM981	Niños del mar
PFM378	Disculpe!
PFM378	Green Cold Soup Beans
PFM379	Spice Gals
PFM451	Galifianakis
PFM451	The Freaks
PFM534	Obrigados
PFM437	I Gatti
PFM480	Obrigados
PFM591	Les Baguettes
PFM709	La revolução
PFM322	La revolução
PFM545	Los Enanitos
PFM423	Foo Allies
PFM801	Los Enanitos

Figure 8. performance_artist (M:N) table (portion)

stand_name	festival_name	estival_start_dat	festival_location	stand_sponsor
Tune Treats	Velvet Noise	2025-09-04	Tokyo	Altura Foods
BeatStreet Stanc	Midnight Grove	2024-02-23	Berlin	Zenith Apparel
Merch & Munch	Electric Valley	2024-01-16	Monaco	Solaris Entertainment
SnackStage	Electric Valley	2024-01-16	Monaco	IronLeaf Brewing Co.
BeatStreet Stanc	Neon Pulse	2025-04-19	Milan	NovaCorp Industries
Rock 'n' Roll Refi	Aurora Soundwa	2024-03-07	Milan	Zenith Apparel
SoundBite Corne	Sunset Echo	2025-07-25	Berlin	Vortex Digital Media
Harmony Hut	Neon Pulse	2024-12-27	Monaco	Altura Foods
Tune Treats	Sunset Echo	2025-05-28	Napels	Polyphonic Records
Echo Merch & M	Desert Mirage	2024-08-30	Philadelphia	NovaCorp Industries
Vibe & Sip	Desert Mirage	2025-01-10	Tokyo	VelvetSky Cosmetics
Tune Treats	Cloudstage	2025-12-06	Milan	Vortex Digital Media
Vibe & Sip	Electric Valley	2024-11-12	Barcellona	Zenith Apparel
BeatStreet Stanc	Northern Lights	2025-08-13	Monaco	QuantumSound Systems
Rock 'n' Roll Refi	Rhythm Horizon	2024-06-20	Napels	VelvetSky Cosmetics
Echo Merch & M	Northern Lights	2024-09-24	Rome	QuantumSound Systems
Encore Eats	Velvet Noise	2025-06-18	Barcellona	Lumen Energy Drinks
SnackStage	Desert Mirage	2024-08-30	Philadelphia	Altura Foods
Vibe & Sip	Desert Mirage	2024-08-30	Philadelphia	EchoStream Media
Merch & Munch	Velvet Noise	2025-11-09	Napels	Polyphonic Records
SoundBite Corne	Desert Mirage	2024-08-30	Philadelphia	Solaris Entertainment
Groove & Brew	Sunset Echo	2025-05-28	Napels	Polyphonic Records
Rhythm Refresh	Desert Mirage	2024-08-30	Philadelphia	Polyphonic Records
Rhythm Refresh	Velvet Noise	2025-11-09	Napels	Orion Tech Labs
SoundBite Corne	Rhythm Horizon	2025-03-02	Rome	WaveLine Studios
BeatStreet Stanc	Northern Lights	2024-10-09	Madrid	AeroVista Travel
Rhythm Refresh	Rhythm Horizon	2024-06-20	Napels	Orion Tech Labs
Vibe & Sip	Northern Lights	2025-08-13	Monaco	Lumen Energy Drinks
Groove Gear & (	Rhythm Horizon	2025-03-02	Rome	NeuroPulse Technologies
Melody Merch &	Velvet Noise	2025-06-18	Barcellona	Zenith Apparel
Groove & Brew	Northern Lights	2024-09-24	Rome	EchoStream Media
Harmony Hut	Sunset Echo	2024-04-21	Tokyo	Polyphonic Records
SnackStage	Sunset Echo	2025-05-28	Napels	Nimbus Event Group
SoundBite Corne	Aurora Soundwa	2025-09-10	Milan	CrystalBay Beverages
Rock 'n' Roll Refi	Velvet Noise	2025-11-09	Napels	Zenith Apparel
Amplify Eats	Rhythm Horizon	2024-02-11	Rome	AeroVista Travel
Sonic Snack	Velvet Noise	2025-11-09	Napels	AeroVista Travel
StageFuel	Aurora Soundwa	2024-03-07	Milan	Vortex Digital Media
BeatBites	Northern Lights	2025-08-13	Monaco	WaveLine Studios
Rock 'n' Roll Refi	Desert Mirage	2024-05-15	Milan	Altura Foods
Rock 'n' Roll Refi	Sunset Echo	2025-05-28	Napels	Polyphonic Records
Tune Treats	Aurora Soundwa	2025-09-10	Milan	Orion Tech Labs
BeatStreet Stanc	Aurora Soundwa	2024-03-07	Milan	WaveLine Studios
Tune Treats	Desert Mirage	2024-08-30	Philadelphia	Vortex Digital Media
Encore Eats	Rhythm Horizon	2024-02-11	Rome	Altura Foods
Groove & Brew	Velvet Noise	2025-09-04	Tokyo	Lumen Energy Drinks

Figure 9. stand table (portion)

product_name	product_price
Mineral Water Still 500m	1,5
Sparkling Water	1,5
Apple Juice 300ml	2,3
Orange Juice Fresh 300m	2,5
Pineapple Juice 300ml	2,5
Watermelon Juice 300ml	2,6
Cranberry Juice 300ml	2,8
Coca Cola 330ml	2
Diet Cola 330ml	2
Grape Soda 330ml	2,1
Ginger Ale 330ml	2
Sparkling Lemon 300ml	1,8
Coconut Water 350ml	2,9
Chai Latte Cold 350ml	3,5
Iced Tea Peach 300ml	2,2
Iced Tea Lemon 300ml	2,2
Iced Green Tea 300ml	2,2
Sprite 500ml	3
Fanta Orange 330ml	2
Mountain Dew 500ml	5
Red Bull 330ml	3,5
Pepsi 300ml	2
Diet Pepsi Max 300ml	2
7 Up 300ml	2
7 Up Zero Sugar 300ml	2
Classic Burger	8
Cheeseburger	8,5
Bacon Burger	9
Vegan Burger	8,5
Pulled Pork Sandwich	9
Fried Chicken Sandwich	8
Grilled Cheese	6,5
Hot Dog	6
Chicken Wrap	7,5
Nachos with Cheese	6,5
Loaded Fries	6,5
Chicken Nuggets	6
Onion Rings	5
Pizza Slice	5
Club Sandwich	7,5
Ham and Cheese Toast	6
Caprese Salad	7
Cold Cuts Plate	8,5
Tuna Salad	6,5
Salted Peanuts	2,5
Hummus with Bread	5,5

Figure 10. product table (portion)

product_name	drink_exp_date
Mineral Water Still 500ml	2028-02-09
Sparkling Water	2030-10-18
Apple Juice 300ml	2029-11-13
Orange Juice Fresh 300ml	2029-12-21
Pineapple Juice 300ml	2030-11-13
Watermelon Juice 300ml	2028-06-09
Cranberry Juice 300ml	2029-02-05
Coca Cola 330ml	2028-11-08
Diet Cola 330ml	2030-03-20
Grape Soda 330ml	2028-03-24
Ginger Ale 330ml	2030-03-15
Sparkling Lemon 300ml	2030-08-22
Coconut Water 350ml	2028-05-04
Chai Latte Cold 350ml	2028-05-04
Iced Tea Peach 300ml	2030-01-21
Iced Tea Lemon 300ml	2030-09-04
Iced Green Tea 300ml	2029-02-07
Sprite 500ml	2030-10-06
Fanta Orange 330ml	2028-01-03
Mountain Dew 500ml	2029-01-18
Red Bull 330ml	2030-01-04
Pepsi 300ml	2029-03-05
Diet Pepsi Max 300ml	2030-02-15
7 Up 300ml	2029-08-06
7 Up Zero Sugar 300ml	2029-05-12

Figure 11. product_drink table

product_name	food_exp_date	food_type
Potato Chips 15	2027-03-14	salty
Cheese Cracker	2028-08-09	salty
Bacon Flavored	2028-01-22	salty
Vegan Granola	2027-05-03	vegan
Mixed Nuts Sna	2028-10-17	vegan
Pretzel Sticks 1(	2027-09-08	salty
Chocolate Muff	2028-07-28	sweet
Mini Hot Dog-Fl	2028-06-06	salty
Chicken Flavore	2027-04-15	salty
Nacho Cheese E	2028-11-30	salty
Salted Popcorn	2027-09-21	vegan
BBQ Corn Snack	2028-02-04	salty
Onion Rings Sn:	2027-10-10	vegetarian
Mini Pizza Crac	2028-01-05	salty
Sandwich Cooki	2027-07-09	sweet
Cheese Balls 80	2028-12-02	salty
Tomato Breads	2028-06-26	vegetarian
Salami Snack St	2027-08-12	meat-based
Tuna Flavored (	2028-09-11	meat-based
Roasted Peanut	2027-03-22	vegan
Hummus Chips	2028-04-17	vegan
Cotton Candy C	2027-02-09	sweet
Chocolate Don	2028-05-25	sweet
Fruit Gummies	2027-08-13	sweet
Ice Cream Sand	2028-12-01	sweet

Figure 12. product_food table

product_name
T Shirt Logo Band
Hoodie Band Design
Hoodie Festival Design
Mug with Logo
Poster 60x90 cm
Sticker Pack 10 pcs
Sticker pack 20 pcs
Tote Bag Logo
Keychain
Beanie Hat
Socks
Festival Pen
Pin Badge
Wristband Silicone
Poster Limited Edition
Coffee Table Book
Guitar Pick Set
Drumstick Set Replica
Phone Case Band Logo
Sticker Vinyl Large
Concert Program Booklet
Limited Edition Poster
Collector Pin Set
Sunglasses
Baseball Cap

Figure 13. product_merchandise table

stand_name	festival_name	festival_start_date	festival_location	product_name
SoundBite Corner	Electric Valley	2024-01-16	Monaco	Radler Can 500ml
Sonic Snack	Velvet Noise	2025-11-09	Naples	Mug with Logo
Melody Merch & M	Velvet Noise	2025-06-18	Barcellona	Donut
SoundBite Corner	Rhythm Horizon	2025-03-02	Rome	Vegan Burger
Encore Eats	Northern Lights	2025-08-13	Monaco	Cranberry Juice 300ml
Tune Treats	Velvet Noise	2025-11-09	Naples	Cotton Candy
Sonic Snack	Sunset Echo	2024-04-21	Tokyo	Chicken Wrap
Rock 'n' Roll Refresl	Velvet Noise	2025-11-09	Naples	Baseball Cap
Encore Eats	Midnight Grove	2024-02-23	Berlin	Coca Cola 330ml
Tune Treats	Velvet Noise	2025-11-09	Naples	Iced Tea Peach 400ml
Rock 'n' Roll Refresl	Aurora Soundwave	2024-03-07	Milan	Wristband Silicone
Festival Feast	Sunset Echo	2025-05-28	Naples	Tuna Salad
Amplify Eats	Electric Valley	2024-11-12	Barcellona	Sticker pack 20 pcs
Bassline Bites	Electric Valley	2024-07-01	Madrid	Sticker Vinyl Large
Sonic Snack	Velvet Noise	2025-11-09	Naples	Lager Draft Pint
BeatBites	Northern Lights	2025-08-13	Monaco	Ham and Cheese Toast
Bassline Bites	Rhythm Horizon	2024-02-11	Rome	Drumstick Set Replica
Rhythm Refreshmei	Desert Mirage	2024-08-30	Philadelphia	Sunglasses
Encore Eats	Northern Lights	2025-08-13	Monaco	Orange Juice Fresh 300ml
Amplify Eats	Electric Valley	2024-11-12	Barcellona	Cranberry Juice 300ml
SnackStage	Desert Mirage	2024-08-30	Philadelphia	Watermelon Juice 300ml
Amplify Eats	Rhythm Horizon	2024-02-11	Rome	Amber Ale Cup 400ml
Rock 'n' Roll Refresl	Aurora Soundwave	2024-03-07	Milan	Ice Cream Cup
BeatStreet Stand	Midnight Grove	2024-02-23	Berlin	Socks
Echo Merch & Mun	Velvet Noise	2025-11-09	Naples	Hoodie Festival Design
SoundBite Corner	Rhythm Horizon	2025-03-02	Rome	Apple Juice 300ml
Merch & Munch Sp	Velvet Noise	2025-11-09	Naples	Diet Cola 330ml
Echo Merch & Mun	Northern Lights	2024-09-24	Rome	Orange Juice Fresh 300ml
BeatStreet Stand	Rhythm Horizon	2024-02-11	Rome	Poster Limited Edition
Melody Merch & M	Velvet Noise	2025-06-18	Barcellona	Salted Peanuts
Amplify Eats	Northern Lights	2024-09-24	Rome	Onion Rings
Groove Gear & Grul	Electric Valley	2024-11-12	Barcellona	Watermelon Juice 300ml
Groove & Brew	Sunset Echo	2024-04-21	Tokyo	Hoodie Band Design
Groove & Brew	Northern Lights	2024-09-24	Rome	Watermelon Juice 300ml
SoundBite Corner	Desert Mirage	2024-08-30	Philadelphia	Pulled Pork Sandwich
Festival Feast	Rhythm Horizon	2024-02-11	Rome	Green Tea Cold Brew 350ml
Amplify Eats	Midnight Grove	2024-02-23	Berlin	Limited Edition Poster
Groove & Brew	Electric Valley	2024-01-16	Monaco	Donut
Harmony Hut	Neon Pulse	2024-12-27	Monaco	IPA Craft Can 440ml
Harmony Hut	Sunset Echo	2024-04-21	Tokyo	Socks
Harmony Hut	Electric Valley	2024-01-16	Monaco	Tuna Salad
SnackStage	Electric Valley	2024-01-16	Monaco	Cotton Candy
Groove Gear & Grul	Velvet Noise	2025-09-04	Tokyo	Iced Tea Peach 400ml
SoundBite Corner	Aurora Soundwave	2025-09-10	Milan	Hoodie Band Design
Harmony Hut	Velvet Noise	2025-11-09	Naples	T Shirt Logo Band
Harmony Hut	Velvet Noise	2025-06-18	Barcellona	Nachos with Cheese

Figure 14. stand_product (M:N) table (portion)

person_email_address	person_full_name	person_phone_number
mdownage0@usa.gov	Mattheus Downage	129-885-9594
bvowden1@nhs.uk	Blair Vowden	380-981-2636
ahabishaw2@shareasale.co	Antoni Habishaw	937-195-0973
pcressar3@va.gov	Pepito Cressar	235-332-0899
jpendall4@fastcompany.co	Jonie Pendall	512-143-4324
anewart5@etsy.com	Ainslee Newart	509-549-9973
ienrietto6@google.ru	Iago Enrietto	506-305-6090
kmary7@bluehost.com	Kimmi Mary	201-924-5331
tletherbury8@cdc.gov	Terrie Letherbury	469-452-8170
mbergin9@dailymail.co.uk	Monica Bergin	372-447-6724
tkearlea@elegantthemes.c	Timmie Kearle	701-514-1348
wberndtssenb@unblog.fr	Walther Berndtssen	972-195-5818
maldisc@gnu.org	Malina Aldis	151-317-7722
ggenteryd@geocities.com	Gabby Gentery	358-896-3272
bofiellye@ftc.gov	Boigie O'Fielly	700-839-8696
rpeacockef@businesswire.c	Ransom Peacocke	220-976-2895
ehinkseng@jigsy.com	Edy Hinksen	855-760-8938
gdargavelh@mayoclinic.cor	Gail Dargavel	299-313-5638
slochheadi@cafepress.com	Sela Lochhead	470-946-5517
eboissierj@tmall.com	Erminia Boissier	364-613-5186
vbullickk@ameblo.jp	Valaria Bullick	465-772-4738
amcgrudderl@marriott.com	Avie McGrudder	570-873-1601
aivorym@4shared.com	Adore Ivory	185-173-9907
fharringtonn@quantcast.co	Fabian Harrington	618-884-1829
ftrillowo@themeforest.net	Fons Trillow	421-325-2866
elowisp@sfgate.com	Eimile Lowis	888-590-3517
fhartshornq@adobe.com	Francine Hartshorn	688-511-5793
jjohananoffr@businessinsid	Joel Johananoff	722-164-5939
oskatess@eventbrite.com	Olimpia Skates	317-506-3314
bcrucittit@arstechnica.com	Blondy Crucitti	115-776-6476
cwasbroughu@blogspot.co	Clo Wasbrough	185-149-1910
ccarsonv@ft.com	Carleton Carson	835-949-2018
jeglesew@rediff.com	Jerald Eglese	189-185-3564
nbantockx@yellowpages.cc	Nevsa Bantock	603-258-0311
rfullmany@blog.com	Rourke Fullman	475-180-9052
aoldfieldcherryz@cmu.edu	Arlie Oldfield-Cherry	422-170-9147
ccapey10@marketwatch.cc	Corbet Capey	810-546-7486
gheymann11@canalblog.co	Gary Heymann	419-883-1478
do12@globo.com	Darn O' Mullane	942-958-1457
sgamwell13@tinyurl.com	Sarina Gamwell	936-971-1465
fburge14@g.co	Ferdy Burge	348-887-9364
dbuller15@netlog.com	Dav Buller	392-980-6483
hmanifould16@disqus.com	Hayyim Manifould	758-687-8360
castlet17@dyndns.org	Cassandra Astlet	729-749-1187
ckingswoode18@auda.org.a	Coraline Kingswoode	356-619-9041
bboss19@reuters.com	Bink Boss	150-139-9346

Figure 15. person table (portion)

purchase_id	on_email_adc	urchase_dat	se_payment_method
P-381729	kaird57@fotk	2025-08-04	CARD
P-472015	amaud6o@tr	2023-12-02	ONLINE
P-938240	cjoplin3z@va	2024-02-03	ONLINE
P-127643	wlightbowne	2024-05-06	OTHER
P-508392	akees5t@unb	2024-05-15	ONLINE
P-762918	erapelli2h@f	2024-12-12	CARD
P-690451	mdyshart4l@	2024-12-01	CASH
P-845239	amcgrudderl	2023-12-22	CARD
P-613708	fhendrickson	2025-01-03	ONLINE
P-572491	rspyvye6v@s	2025-04-05	CASH
P-904672	bdmitrienko2	2025-10-04	OTHER
P-384107	ccordingly2u	2025-09-11	CASH
P-147980	hhubback2b@	2025-01-07	ONLINE
P-218635	lbreckell3a@	2024-04-15	ONLINE
P-845016	gfarnham67@	2024-03-07	OTHER
P-912843	fgatchell7r@	2024-05-09	CARD
P-634892	brishworth2y	2024-07-31	ONLINE
P-372415	cpardi73@ba	2024-08-28	ONLINE
P-958102	mpenas33@e	2025-06-01	CARD
P-742650	jjerdein6u@g	2025-12-05	CASH
P-836219	ktremble7z@	2024-11-29	ONLINE
P-601347	cclarage1o@	2024-06-24	ONLINE
P-279534	sgirod7a@las	2024-05-20	ONLINE
P-985143	lwheatley5u@	2025-09-16	ONLINE
P-410627	mmullen3o@	2024-01-07	OTHER
P-578301	miowarch52@	2025-11-28	ONLINE
P-394862	mhaythorne5	2024-12-15	CARD
P-962437	kdomone7e@	2023-12-16	OTHER
P-703821	jrobke2v@tu	2024-02-24	ONLINE
P-651084	strowl3q@biz	2025-08-12	ONLINE
P-829765	rloughlin5k@	2025-08-11	ONLINE
P-138249	mwilcher7g@	2024-05-14	CARD
P-570934	ggenteryd@g	2024-05-20	ONLINE
P-461293	sbothe4o@re	2025-08-29	ONLINE
P-218709	tkearlea@ele	2024-09-17	ONLINE
P-963842	fhartshornq@	2024-09-05	ONLINE
P-804152	dfelix5f@the	2025-07-02	ONLINE
P-659471	jcattlow20@c	2024-08-10	ONLINE
P-973256	lsterley58@a	2025-09-23	ONLINE
P-390517	rpiercy4m@v	2024-04-20	ONLINE
P-547918	eboissierj@tr	2025-08-05	CASH
P-416208	bmafham45@	2023-12-17	ONLINE
P-291870	mbrunsden7(	2024-11-17	ONLINE
P-781643	grabjohns1w	2024-11-06	ONLINE
P-605982	lloblie5m@gc	2025-07-10	CARD
P-932417	fharringtonn	2023-12-19	ONLINE

Figure 16. purchase table (portion)

ticket_id	ticket_type	icket_pric	t_validity_	t_validity_	stival_nam	ival_start_	tival_locat	urchase_id
TCK1224	DAY-PASS	10	2025-09-1	2025-09-1	Aurora Sou	2025-09-1	Milan	P-381729
TCK1225	DAY-PASS	10	2024-01-1	2024-01-1	Electric Va	2024-01-1	Monaco	P-472015
TCK1226	DAY-PASS	10	2024-02-1	2024-02-1	Rhythm Hc	2024-02-1	Rome	P-938240
TCK1227	DAY-PASS	10	2024-06-2	2024-06-3	Rhythm Hc	2024-06-2	Naples	P-127643
TCK1228	DAY-PASS	10	2024-07-0	2024-07-0	Electric Va	2024-07-0	Madrid	P-508392
TCK1229	DAY-PASS	10	2025-01-0	2025-01-0	Neon Pulse	2024-12-2	Monaco	P-762918
TCK1230	DAY-PASS	10	2025-01-1	2025-01-2	Desert Mir	2025-01-1	Tokyo	P-690451
TCK1231	DAY-PASS	10	2024-02-1	2024-02-1	Rhythm Hc	2024-02-1	Rome	P-845239
TCK1232	DAY-PASS	10	2025-01-2	2025-01-2	Desert Mir	2025-01-1	Tokyo	P-613708
TCK1233	DAY-PASS	10	2025-04-2	2025-04-2	Neon Pulse	2025-04-1	Milan	P-572491
TCK1234	DAY-PASS	10	2025-11-1	2025-11-1	Velvet Noi	2025-11-0	Naples	P-904672
TCK1235	DAY-PASS	10	2025-10-0	2025-10-0	Midnight C	2025-10-0	Rome	P-384107
TCK1236	DAY-PASS	10	2025-01-2	2025-01-2	Desert Mir	2025-01-1	Tokyo	P-147980
TCK1237	DAY-PASS	10	2024-05-0	2024-05-0	Sunset Ech	2024-04-2	Tokyo	P-218635
TCK1238	DAY-PASS	10	2024-05-0	2024-05-0	Sunset Ech	2024-04-2	Tokyo	P-845016
TCK1239	DAY-PASS	10	2024-07-0	2024-07-0	Electric Va	2024-07-0	Madrid	P-912843
TCK1240	DAY-PASS	10	2024-09-1	2024-09-1	Desert Mir	2024-08-3	Philadelph	P-634892
TCK1241	DAY-PASS	10	2024-10-1	2024-10-1	Northern L	2024-10-0	Madrid	P-372415
TCK1242	DAY-PASS	10	2025-06-2	2025-06-2	Velvet Noi	2025-06-1	Barcellona	P-958102
TCK1243	DAY-PASS	10	2025-12-1	2025-12-1	Cloudstage	2025-12-0	Milan	P-742650
TCK1244	DAY-PASS	10	2024-12-2	2024-12-2	Neon Pulse	2024-12-2	Monaco	P-836219
TCK1245	DAY-PASS	10	2024-06-2	2024-06-3	Rhythm Hc	2024-06-2	Naples	P-601347
TCK1246	DAY-PASS	10	2024-06-2	2024-06-2	Rhythm Hc	2024-06-2	Naples	P-279534
TCK1247	DAY-PASS	10	2025-10-0	2025-10-1	Midnight C	2025-10-0	Rome	P-985143
TCK1248	DAY-PASS	10	2024-02-2	2024-02-2	Midnight C	2024-02-2	Berlin	P-410627
TCK1249	DAY-PASS	10	2025-12-1	2025-12-1	Cloudstage	2025-12-0	Milan	P-578301
TCK1250	DAY-PASS	10	2025-01-0	2025-01-0	Neon Pulse	2024-12-2	Monaco	P-394862
TCK1251	DAY-PASS	10	2024-02-1	2024-02-1	Rhythm Hc	2024-02-1	Rome	P-962437
TCK1252	DAY-PASS	10	2024-03-0	2024-03-1	Aurora Sou	2024-03-0	Milan	P-703821
TCK1253	DAY-PASS	10	2025-10-0	2025-10-0	Midnight C	2025-10-0	Rome	P-651084
TCK1254	DAY-PASS	10	2025-09-1	2025-09-1	Velvet Noi	2025-09-0	Tokyo	P-829765
TCK1255	DAY-PASS	10	2024-06-2	2024-06-2	Rhythm Hc	2024-06-2	Naples	P-138249
TCK1256	DAY-PASS	10	2024-06-2	2024-06-2	Rhythm Hc	2024-06-2	Naples	P-570934
TCK1257	DAY-PASS	10	2025-09-1	2025-09-1	Velvet Noi	2025-09-0	Tokyo	P-461293
TCK1258	DAY-PASS	10	2024-10-1	2024-10-1	Northern L	2024-10-0	Madrid	P-218709
TCK1259	DAY-PASS	10	2024-10-1	2024-10-1	Northern L	2024-10-0	Madrid	P-963842
TCK1260	DAY-PASS	10	2025-08-2	2025-08-2	Northern L	2025-08-1	Monaco	P-804152
TCK1261	DAY-PASS	10	2024-09-2	2024-09-2	Northern L	2024-09-2	Rome	P-659471
TCK1262	DAY-PASS	10	2025-11-1	2025-11-1	Velvet Noi	2025-11-0	Naples	P-973256
TCK1263	DAY-PASS	10	2024-04-2	2024-04-2	Sunset Ech	2024-04-2	Tokyo	P-390517
TCK1264	DAY-PASS	10	2025-09-1	2025-09-1	Velvet Noi	2025-09-0	Tokyo	P-547918
TCK1265	DAY-PASS	10	2024-02-1	2024-02-1	Rhythm Hc	2024-02-1	Rome	P-416208
TCK1266	DAY-PASS	10	2024-11-2	2024-11-2	Electric Va	2024-11-1	Barcellona	P-291870
TCK1267	DAY-PASS	10	2024-12-2	2024-12-3	Neon Pulse	2024-12-2	Monaco	P-781643
TCK1268	DAY-PASS	10	2025-08-1	2025-08-2	Northern L	2025-08-1	Monaco	P-605982
TCK1269	DAY-PASS	10	2024-01-1	2024-01-2	Electric Va	2024-01-1	Monaco	P-932417

Figure 17. ticket table (portion)

product_name	purchase_id
Loaded Fries	P-385055
Nachos with Cheese	P-975673
Coconut Water 350ml	P-147369
Sticker pack 20 pcs	P-458655
Socks	P-706338
Hummus with Bread	P-401706
Poster Limited Edition	P-698025
Coca Cola 330ml	P-266194
Socks	P-687017
Red Bull 330ml	P-421812
Apple Juice 300ml	P-757401
Ice Cream Cup	P-654175
Ginger Ale 330ml	P-175293
Coffee Table Book	P-471345
Club Sandwich	P-869894
Donut	P-320552
Keychain	P-482591
Sunglasses	P-362803
Red Bull 330ml	P-229241
7 Up 300ml	P-136297
Pizza Slice	P-563474
Fruit Popsicle	P-319273
Donut	P-394735
Sparkling Water	P-633472
Ginger Ale 330ml	P-261212
Bacon Burger	P-502909
Sparkling Lemon 300ml	P-874996
Pineapple Juice 300ml	P-990980
Diet Pepsi Max 300ml	P-696486
Pizza Slice	P-365209
Grilled Cheese	P-556632
Loaded Fries	P-361277
Iced Green Tea 300ml	P-985657
Diet Pepsi Max 300ml	P-331983
Guitar Pick Set	P-812072
Grilled Cheese	P-129878
Pulled Pork Sandwich	P-344192
Grilled Cheese	P-583205
Concert Program Bookle	P-166451
Loaded Fries	P-925668
Hoodie Festival Design	P-159551
Limited Edition Poster	P-607580
Onion Rings	P-851155
Salted Peanuts	P-417776
Cold Cuts Plate	P-848044
Watermelon Juice 300m	P-558523

Figure 18. product_purchase (M:N) table (portion)

6 | SQL Queries

6.1. WHERE – "Performances that take place at Naples"

SELECT performance_unique_id, performance_title, performance_genre, performance_start_time
FROM performance
WHERE festival_location = 'Naples'

Query description:
The selected performance data (i.e., the columns 'performance_unique_id', 'performance_title', 'performance_genre', 'performance_start_time') is retrieved from the 'performance' table and then the results are restricted by location equal to 'Naples' with 'WHERE'.

Query results:

formance_unique	erformance_titl	erformance_genr	formance_start_time
PFM565	Los Enanitos Indi	Indie	2025-06-01 20:00
PFM198	Puh R&B Session	R&B	2025-06-04 21:00
PFM158	La revolução Indi	Indie	2025-05-28 21:00
PFM775	The Cockroaches	Pop	2025-06-03 14:45
PFM498	KitKats Live Electı	Electronic	2024-06-24 18:45
PFM629	KitKats – Electron	Electronic	2024-06-23 15:15
PFM801	Los Enanitos Indi	Indie	2024-06-25 14:00
PFM316	Redday Live Regg	Reggae	2024-06-24 18:30
PFM697	Les Baguettes Liv	Folk	2025-11-10 20:15
PFM878	Blancodolce Live	Electronic	2025-11-12 19:45
PFM251	Redday Reggae S	Reggae	2025-05-28 14:45
PFM448	I Gatti Pop Sessio	Pop	2025-06-01 16:15
PFM334	Foo Allies Live Co	Country	2025-11-14 22:30
PFM581	Galifianakis Jazz S	Jazz	2025-11-15 17:30
PFM915	Les Baguettes Liv	Folk	2025-11-09 17:30
PFM465	I Gatti – Pop Set	Pop	2025-11-17 14:00
PFM437	I Gatti Live Pop	Pop	2025-11-13 19:00
PFM544	Zeb&friends Live	House	2025-11-17 20:45
PFM666	Disculpe! Live Fol	Folk	2025-11-12 14:30
PFM610	Blancodolce Elect	Electronic	2025-06-07 15:45
PFM505	Disculpe! – Folk S	Folk	2024-06-20 14:15
PFM133	BimBamBum – M	Metal	2025-06-05 15:15
PFM500	Galifianakis Jazz S	Jazz	2024-06-22 13:00
PFM607	The Cockroaches	Pop	2024-07-01 12:30
PFM784	BimBamBum Met	Metal	2025-11-12 15:00
PFM371	Galifianakis Jazz S	Jazz	2024-06-21 21:15

Figure 19. Query 'WHERE' output

6.2. WHERE, LIMIT, LIKE – "Personal info of a musician whose name starts with 'D' "

```
SELECT artist_stage_name, musician_full_name, musician_nationality, musician_main_instrument
FROM artist_musician
WHERE artist_stage_name LIKE 'D%'
LIMIT 1;
```

Query description:
The 'WHERE' clause filters the rows in the 'artist_musician' table using the pattern `'D%'`, where '%' allows for any characters following 'D'.
The 'LIMIT 1' clause restricts the output to only one matching musician.
Without an 'ORDER BY', the specific musician returned may vary depending on how the database retrieves rows.

Query results:

artist_stage_name	musician_full_name	musician_nationality	usician_main_instrument
Duplo	Sonia Hummels	German	Drums

Figure 20. Query 'WHERE, LIMIT, LIKE' output

6.3. WHERE, IN, Nested Query – "Distinct products sold at the festival Sunset Echo Tokyo started on 2024-04-21"

```
SELECT DISTINCT stand_product, product_name
FROM stand_product
WHERE (stand_product, festival_name, stand_product, festival_start_date, stand_product,
festival_location) IN (
    SELECT festival_name, festival_start_date, festival_location
    FROM festival
    WHERE festival_name = 'Sunset Echo' AND festival_start_date = '2024-04-21'
);
```

Query description:
The inner query selects the specific festival edition of Sunset Echo held on 2024-04-21

(identified by the combination of name, start date, and location).
The outer query retrieves all distinct product names from the 'stand_product' table that are linked to that same festival edition.
The 'DISTINCT' keyword ensures that each product type appears only once in the result list, even if it is sold at multiple stands.

Query results:

product_name
Chicken Wrap
Hoodie Band Design
Socks
Collector Pin Set
Orange Juice Fresh 300ml
Sticker Pack 10 pcs
Iced Tea Peach 400ml
Drumstick Set Replica
Hot Dog
Caprese Salad
Lager Draft Pint
IPA Craft Can 440ml
Concert Program Booklet
Pineapple Juice 300ml
Diet Cola 330ml
Loaded Fries
Beanie Hat
Ice Cream Cup
Pizza Slice
Pin Badge
Phone Case Band Logo
Cranberry Juice 300ml
Club Sandwich
Baseball Cap
Sunglasses
Apple Juice 300ml
Tuna Salad
Classic Burger
Keychain
Spritz Cup
Mineral Water Still 500ml
Pulled Pork Sandwich
Sparkling Lemon 300ml
Coconut Water 350ml
Wristband Silicone
Cotton Candy
Hoodie Festival Design
Limited Edition Poster
Lager Draft Half Pint
Chai Latte Cold 350ml
Onion Rings
Ginger Ale 330ml
Fried Chicken Sandwich

Figure 21. Query WHERE, IN, Nested Query output

6.4. GROUP BY, 1 JOIN, AS – "Number of performances per stage across all festivals"

```
SELECT s.stage_name AS stage, COUNT(p.performance_unique_id) AS num_performances
FROM stage s
JOIN performance p
    ON p.festival_name = s.festival_name
    AND p.festival_start_date = s.festival_start_date
    AND p.festival_location = s.festival_location
    AND p.stage_identification_code = s.stage_identification_code
GROUP BY s.stage_name
ORDER BY num_performances DESC;
```

Query description:

At first, aliases are created with 'AS' (stage names as 'stage' and the number of performances counted with COUNT as 'num_performances'). Then the table stage is retrieved with 'FROM' and assigned the alias 's'. Then it is joined with the table 'performance' (with its respective alias 'p') using the 'JOIN' function. Each performance is linked with its stage by matching the primary key of stage with the respective foreign key of performance. 'GROUP BY' groups the rows based on stage names: hence, for each stage name the number of performances that take place there are shown. 'ORDER BY' helps to sort in descending ('DESC') order the data retrieved, to show the most used stages first.

Query results:

stage	num_performances
Main	24
Beam	21
Forge	19
Stage A	17
Echo	17
Peak	16
Bloom	16
Edge	14
Drift	14
Nexus	13
Vibe	12
Roots	12
Haze	12
Axis	12
Rift	9
Yard	8
Vista	8
Stone	8
Pulse	8
Arc	8
Core	7
Zen	6
Shift	6
Luna	6
Harbor	6
Field	6
Solar	5
Stage C	4
Crest	4
Nova	2

Figure 22. Query GROUP BY, 1 JOIN, AS output

6.5. WHERE, GROUP BY – "Tickets sold by type at the festival Velvet Noise Barcellona started on 2025-06-18"

```
SELECT ticket_type, COUNT(*) AS tickets_sold
FROM ticket
WHERE festival_name = 'Velvet Noise'
AND festival_start_date = '2025-06-18'
AND festival_location = 'Barcellona'
GROUP BY ticket_type;
```

Query description:

The query retrieves ticket sales information from the 'ticket' table. The 'WHERE' clause filters the rows to include only tickets belonging to the Velvet Noise festival edition in Barcellona (2025-06-18). The 'GROUP BY ticket_type' groups the filtered records by ticket category (Day Pass, Regular, VIP). The 'COUNT(*)' function then counts how many tickets of each type were sold, producing one row per ticket category with the corresponding total number of sales.

Query results:

ticket_type	tickets_sold
DAY-PASS	5
REGULAR	4
VIP	4

Figure 23. Query WHERE, GROUP BY output

6.6. GROUP BY, HAVING, AS – "Products purchased at least ten times"

```
SELECT product_name AS product, COUNT(*) AS times_bought
FROM product_purchase
GROUP BY product_name
HAVING COUNT(*) >= 10
ORDER BY times_bought DESC, product;
```

Query description:

The query retrieves product names (giving the alias 'product'), and their quantity purchased (giving the alias 'times_bought') with 'COUNT'. It then groups purchase records by product name and counts how many times each product was bought. 'HAVING' with '>=10' filters out products purchased one time, while 'ORDER BY' lists them by popularity. Results are ordered in descending order by number of purchases (i.e. 'times_bought'), while if there are any ties they are ordered by alphabetical order.

Query results:

product	times_bought
Grilled Cheese	22
Hoodie Festival Design	17
Apple Juice 300ml	15
Fried Chicken Sandwich	13
Fruit Popsicle	12
Sparkling Lemon 300ml	12
Guitar Pick Set	11
Salted Peanuts	11
Caprese Salad	10
Collector Pin Set	10
Poster Limited Edition	10
Tuna Salad	10
Watermelon Juice 300ml	10

Figure 24 Query GROUP BY, HAVING, AS output

6.7. WHERE, GROUP BY, HAVING, AS – "People who bought at least two products on October 2025"

```
SELECT pu.person_email_address AS buyer, COUNT(pp.product_name) AS products_count
FROM purchase pu
JOIN product_purchase pp ON pp.purchase_id = pu.purchase_id
WHERE pu.purchase_date BETWEEN '2025-10-01' AND '2025-10-31'
GROUP BY pu.person_email_address
HAVING COUNT(pp.product_name) >= 2;
```

Query description:
The query retrieves data from the 'purchase' table, joining it with 'product_purchase' ('pp') through a 'JOIN' on the 'purchase_id' field to link each purchase to its products. The 'WHERE' clause filters purchases made between '2025-10-01' and '2025-10-31', while 'GROUP BY' groups the results by buyer so that 'COUNT(pp.product_name)' counts the number of products per person. The 'HAVING' condition ensures only buyers who bought at least two products appear in the output. Finally, the output lists all buyers who purchased at least one product during October 2025, showing for each their email address (as 'buyer') and the total number of products they bought (as 'products_count').

Query results:

buyer	products_count
sgamwell13@tinyurl.com	2

Figure 25. Query WHERE, GROUP BY, HAVING, AS output

6.8. WHERE, Nested Query, GROUP BY – "Performances per artist in Milan"

```
SELECT pa.artist_stage_name, COUNT(*) AS perf_count
FROM performance_artist pa
WHERE pa.performance_unique_id IN (
    SELECT performance_unique_id
    FROM performance
    WHERE festival_location = 'Milan'
)
GROUP BY pa.artist_stage_name
ORDER BY perf_count DESC, pa.artist_stage_name;
```

Query description:

The query selects the artist's stage name and counts the number of associated performances ('COUNT(*) AS perf_count') from the 'performance_artist' table, which links artists to performances. But first, the subquery inside the 'WHERE' clause retrieves all 'performance_unique_id' values from the performance table where the festival took place in 'Milan'; these IDs are then used to filter the outer query. The results are grouped by artist name so that each artist's total number of performances is counted, and finally ordered by the number of performances in descending order ('DESC'), and alphabetically by artist name for ties. Not null values are guaranteed by design.

Query results:

artist_stage_name	perf_count
Obrigados	9
Eins Zwei Polizei	7
The Freaks	7
Blancodolce	6
Eiffelboys	6
Foo Allies	6
La revolução	6
Los Enanitos	6
Green Cold Soup Beans	5
The Wienerwursts	5
BimBamBum	4
Disculpe!	4
KitKats	4
MunichMonsters	4
Spice Gals	4
I Gatti	3
Niños del mar	3
Zeb&friends	3
Galifianakis	2
Linkin parking lot	2
Puh	2
Redday	2
Μπάστα	2
Les Baguettes	1
The Cockroaches	1

Figure 26. Query WHERE, Nested Query, GROUP BY output

6.9. WHERE, GROUP BY, HAVING, 1 JOIN – "Stands selling multiple products over €5"

```
SELECT sp.stand_name, COUNT(p.product_name) AS num_premium_products
FROM stand_product sp
JOIN product p ON p.product_name = sp.product_name
WHERE p.product_price > 5
GROUP BY sp.stand_name
HAVING COUNT(p.product_name) >= 2;
```

Query description:

The query starts from 'stand_product' (alias 'sp'), the junction that lists which products each stand offers, and joins it to product (alias 'p') on 'product_name' so we can read each item's price. The 'WHERE' clause applies a row-level filter before any aggregation, keeping only those joined rows where 'p.product_price > 5' (the price threshold we're aiming for). Next, 'GROUP BY sp.stand_name' collapses all remaining rows for the same stand into a single group. Within each group, 'COUNT(p.product_name)' counts how many premium products that stand offers; the result is exposed as 'num_premium_products'. Finally, 'HAVING COUNT(p.product_name) >= 2' filters at the group level, ensuring we return only stands that have multiple premium products (i.e. at least two). The output is therefore a list of stands with their corresponding number of premium products, omitting stands that don't meet the price threshold.

Query results:

stand_name	num_premium_products
Amplify Eats	31
Bassline Bites	21
BeatBites	7
BeatStreet Stand	37
Echo Merch & Munch	33
Encore Eats	25
Festival Feast	13
Groove & Brew	32
Groove Gear & Grub	19
Harmony Hut	37
Melody Merch & Munchie	16
Merch & Munch Spot	22
Rhythm Refreshments	16
Rock 'n' Roll Refresh	35
SnackStage	25
Sonic Snack	24
SoundBite Corner	35
StageFuel	8
Tune Treats	37
Vibe & Sip	23

Figure 27. Query WHERE, GROUP BY, HAVING, 1 JOIN output

6.10. WHERE, GROUP BY, HAVING, 2 JOINs – "Festivals that sold at least 5 tickets in cash across all editions"

```
SELECT f.festival_name, COUNT(t.ticket_id) AS tickets_sold
FROM festival f
JOIN ticket t
    ON t.festival_name = f.festival_name
    AND t.festival_start_date = f.festival_start_date
    AND t.festival_location = f.festival_location
JOIN purchase pu
    ON pu.purchase_id = t.purchase_id
WHERE pu.purchase_payment_method = 'CASH'
GROUP BY f.festival_name
HAVING COUNT(t.ticket_id) >= 5
ORDER BY tickets_sold DESC;
```

Query description:

The query starts from the 'festival' table (alias 'f') and performs a first join with 'ticket' (alias 't') to link each festival edition to its tickets, using the shared attributes 'festival_name', 'festival_start_date', and 'festival_location'. Next, it performs a second join with the 'purchase' table (alias 'pu') to associate each ticket with its corresponding purchase record, which contains payment details. The 'WHERE' clause filters the data to include only purchases where the 'purchase_payment_method' equals 'CASH', effectively limiting the results to tickets paid in cash. Then, the 'GROUP BY f.festival_name' groups all remaining tickets by festival name, allowing the aggregate function 'COUNT(t.ticket_id)' to compute how many cash-paid tickets were sold per festival. Finally, the 'HAVING COUNT(t.ticket_id) >= 5' condition ensures that only festivals with at least five cash-paid tickets appear in the output. The 'ORDER BY tickets_sold DESC' helps visualize which festivals sold more tickets in cash. The resulting dataset lists each festival and the number of tickets sold using cash payments.

Query results:

festival_name	tickets_sold
Sunset Echo	13
Desert Mirage	10
Electric Valley	8
Velvet Noise	7
Rhythm Horizon	7
Aurora Soundwave	5

Figure 28. Query WHERE, GROUP BY, HAVING, 2 JOINs output