**BUAN 6320 - Group 4**

**Statement of Work (SOW)**

**1. Overview/Executive Summary**

The objective of this statement of work, is that the work objectives for our customer’s music venue will be outlined clearly, mentions of what items will or won’t be included within the scope of work will be listed, the verified benefits associated with the implementation of the relational database management system (RDBMS), potential future opportunities allowed by the implementation of the relational database management system, and details associated with the above information. This statement of work will serve to describe the uses that will be created by the relational database management system that is implemented, as it’ll be used to streamline current manual processes, which rely heavily on paper records. Due to the nature of the customer’s current venue operations, the implementation of the RDBMS will be to provide efficiencies as well as consistency, in record keeping efforts throughout the business. These record keeping efforts will also consist of customer data gather, historical records of customer purchases, band related information, and data that will assist in the scheduling of performance acts at the business.

**2. Objectives of your database project**

The objective of our database project will be to provide our customer’s venue with a database, of which will hold information of various categories, per the client’s request. These categories will consist of removing current manual record keeping efforts with a relational database management system and to provide an overall database for holding the information from a digital ticketing system user interface, which will be installed by a separate party. Included within the database for the digital ticketing system, there will be a feature for storing ticket information, a data table that will allow ticket information to be parsed by various pricing metrics, a data table for historical recordkeeping of customer information, a data table to hold various order details, a data table to maintain records of concerts being held at the venue, and a data table for storing ticketing information.

**3. Project Scope**

For this work item, our company will be providing the aforementioned items in the objective statement, as well as SQL scripts of the data tables created. These various SQL scripts will provide reproducibility of our efforts, for any future changes or updates. In addition to this, the SQL scripts will cover all data definition language used in the process of creating the client’s database, as well as the data manipulation language used throughout the process.

**In-Scope Work**

• Detailed explanation of database. This will include a noted document describing business rules associated with each table relationship in the database and detailed summaries of attributes within each entity table.

• Entity-relationship model, with descriptions of each attribute. In particular, the client will receive a detailed diagram denoting the table relationships associated with each business entity, through an entity relationship diagram created with Lucidchart™

• (DDL) Data Definition Language Scripts

• (DML) Data Manipulation Language Scripts

**Out-of-Scope Work**

• Anything outside of the above-mentioned items will be considered out of scope

• Consultation work or assisting with the implementation of the digital ticketing system user interface, which is being created by another company, which will eventually utilize the database our company is creating. It is to be agreed upon, that any additional consultative or work apart from the terms listed above, will be brought under a new statement of work or amendment, which will clarify the necessary stipulations and costs to proceed with said work items.

• Additional ad-hoc requests after the aforementioned statement of work-related items have been agreed and signed upon

**4. Database Goals, Expectations, and Deliverables**

Upon completion of the database work items listed above, our customer’s venue will receive an entity relationship diagram depicting 6 entity tables for the agreed to work items. These entity tables will include data holding artist information, concert information, ticketing information, ticket category information, order detail information, and customer information. Each of these entity tables will also be created within Microsoft SQL server. These tables will have relationships created between each of the tables that follow their associated business rules. A document detailing the business rules and relationships between the tables will also be provided. In addition, any SQL scripts that are used in the final creation will be documented and handed over to our customer’s authorized received of said documents. These documents will contain both (DDL) Data Definition Language Scripts and (DML) Data Manipulation Language Scripts, as appropriate for the work task. In addition to this, we’ll also be providing a detailed explanation of how each entity table will contribute to the business’ overall success, based on prior industry or customer experiences.

**5. Database Benefits**

For the business we’re implementing a database for, the usage and potential benefits are several. First, the automation of data entry, using a point-of-sale service provider for ticketing, will help to automate not only the payment processing for the venue, but the venues’ ability to record customer information, which affects the company’s ability to accurately store data. By doing this, the company will be able to keep financial information more accurately, as the human element of handling money is being significantly removed from the equation, as more sales will be conducted via point-of-sale service or via online sales channels. This will then flow through into the database our team is developing, which will get rid of the manual entries on paper sheets, that the venue currently has in place. In addition to this, as customers input their relative data, such as email and purchasing history, the venue will be able to enhance upon its current sales funnels, by increasing consumer awareness of upcoming shows or events that are of value to the consumers themselves, via email promotions. In addition, the venue will also be able to strategically position itself for future use out of the box machine learning algorithms to take advantage of the data being collected. Thereby, the venue will be able to recommend meaningful upcoming concert information or timely discounts, to potential venue consumers in the future, which will ultimately improve their overall profitably as an enterprise

**Project Hardware and Software Tools**

**Diagram Tool**

ER diagram tool by Lucidchart, running on Windows 10

**Office Productivity Tools**

Microsoft Office 365 for Business, running on Windows 10

**Database**

Microsoft SQL Server

**Hardware and Software:**

1. **Hardware requirements**

| Component | Requirement |
| --- | --- |
| Hard Disk | SQL Server requires a minimum of 6 GB of available hard-disk space. |
| Monitor | SQL Server requires Super-VGA (800x600) or higher resolution monitor. |
| Internet | Internet functionality requires Internet access |
| Memory | Minimum:  Express Editions: 512 MB  All other editions: 1 GB |
| Processor Speed | Minimum: x64 Processor: 1.4 GHz  Recommended: 2.0 GHz or faster |
| Processor Type | x64 Processor: AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support |

1. **Software requirements**

| Component | Requirement |
| --- | --- |
| Operating system | Windows 10 or greater |

**Client Access Method**

The client access method will be via SQL Server Management Studio , which they’ll be able to install via the Microsoft website via the following link: https://www.microsoft.com/en-us/sql-server/sql-server-downloads

**7. SQL Usage and Style Guidelines**

* Uppercase for they keywords will be used
* Aliases will be used when they improve readability
* Indentations and white spaces will be used in order to improve readability
* The ANSI-92 JOIN Syntax will be used when joining tables together
* We will use meaningful names based on interpretability
* We will provide comments to indicate what blocks of code accomplish
* Constraint will be given descriptive names
* Overall, we’ll be providing formatting that is aimed at improving readability

**Key Assumptions**

A key business assumption being implemented in the design of the database, is that there’ll be a minimum of one ticket being purchased per concert event, for the concert to occur. Therefore, our database design and ERD diagrams will reflect this assumption, rather than considering the option of using zero to many relationships. The reason for this, is that the business will not put on shows that do not sell any tickets, as it’s a far more inexpensive experience for the business to draw up an agreement to cancel the show with a small deposit fee, than pay the full price of a concert being played to an empty venue.

**Design Decisions**

Key Factors Influencing Design decisions was that it was decided to keep the number of entities (tables) to a manageable number of 6. This was a decision made in part to ensure overcomplexity would not be a problem for this relatively small business venue, but also to help with the overall efficiency of the database. While some entities, such as genre and others were ruled out, other potential entities were combined. For example, the table named “Order Details” was created to take the place of **both** the “Order ticket” and “Customer Order” entities. Therefore, an imperative in our design process was to create something simple, manageable, effective, but overwhelmingly practical for this business to use for years to come.

**Project Management Methodology**

The project management style we approached our project with, was an agile method, where we would meet prior to sprints, which we would use to discuss our plans an iterate on our previous work efforts. As the customers in our database project were fictitious, we would have individuals within our team pretend to be customers, to gain a sense of what a user would want, that wasn’t already being implemented for their business. In conclusion, this gave us an opportunity to think through the various perspectives that customers may have and provided us with more insights into areas we were overlooking pragmatic business needs.

**Business Rules.**   
  
1. An ARTIST may have one or many CONCERTS.   
  
2. A CONCERT may have one or many TICKETS.  
  
3. A CUSTOMER may have one or many ORDERS.

4. An ORDER may have one or many TICKETS associated with it.   
  
5. Every TICKET purchased must have an associated TICKET CATEGORY.

**Relationship and Cardinality Description.**   
  
Relationship: (have) between ARTIST and CONCERT.   
Cardinality: 1:M between ARTIST and CONCERT.   
Business rule: An ARTIST may have one or many CONCERTS and   
each CONCERT held must have an associated and valid ARTIST.   
  
  
Relationship: (sells) between CONCERT and TICKETS.   
Cardinality: 1:M relationship between CONCERT and TICKETS.   
Business rule: A CONCERT may have one or many TICKETS sold but  
each TICKET sold must be linked back to a particular CONCERT.   
  
  
  
Relationship: (has) between TICKET\_CATEGORY and TICKETS.   
Cardinality: 1:1 between TICKET\_CATEGORY and TICKETS.   
Business rule: Each TICKET sold must be associated with a TICKET\_CATEGORY.   
  
  
Relationship: (includes) between ORDER\_DETAILS and TICKETS.   
Cardinality: 1:M between ORDER\_DETAILS and TICKETS.   
Business rule: An ORDER can have one or more TICKETS in it but each ticket must be associated with an ORDER from ORDER\_DETAILS table.

Relationship: (places) between CUSTOMER and ORDER\_DETAILS.   
Cardinality: 1:M between CUSTOMER and ORDER\_ DETAILS.  
Business rule: A CUSTOMER can place one or many ORDERS but each ORDER places must be traced back to at least one CUSTOMER.

**Entity and Attribute Description.**   
  
*Entities*   
  
Entity Name: ARTIST  
  
Entity Description: The ARTIST table describes the list of artists performing at the the customer’s venue. It is connected to the CONCERT entity

Main Attributes of ARTIST:   
artist\_id: (Primary Key) A unique identifier for each artist performing in the concert.   
artist\_name: A character type attribute that informs us about the name of the artist.

genre: a descriptive category denoting the genre of an artist.

genre\_id: an identifier that provides a numerical representation for the genre descriptive category.  
artist\_category: Provides information about the category to which the artist belongs, in regard to their age appropriateness.

artist\_rating: Rating of the artist is used as a numerical representation of the popularity of said artist at the time, as represented by a numerical scale of 1-10

concert\_id: (Foreign key) an attribute that references to CONCERT table.

Entity Name: CONCERT  
  
Entity Description: Next entity of our database is CONCERT, and it stores details about all the concerts. It relates to both ARTIST and TICKETS tables.

Main Attributes of CONCERT:   
concert\_id: (Primary Key) A unique ID for each concert.  
stage\_id: A numeric attribute to uniquely identify various stages in the concert.   
stage\_name: Provides additional information about the stage.

concert\_name: The name of the concert.

concert\_venue: Information about the venue in the concert.

artist\_id: (Foreign key) an attribute that references ARTIST table  
concert\_date: The data where the concert is being held.  
concert\_time: The time at which the concert is being held at.

Entity Name: TICKETS  
  
Entity Description: TICKETS store information about tickets of the concert and is a critical table that connects to 3 other entities; CONCERT, TICKET\_CATEGORY and ORDER\_DETAILS.

Main Attributes of TICKETS:   
ticket\_id: (Primary Key) A unique ID for each ticket.  
serial\_no: The ticket’s serial number, can be utilized as a secondary key for data entry purpose.  
concert\_id: (Foreign key) references the concert table.

purchase\_date: The date when the ticket was purchased.

purchase\_time: The date when the ticket was purchased.

seat\_no: Identifier to uniquely recognize seats at the concert.

tkt\_category\_id: (Foreign key) an attribute that references ticket\_category table.

Entity Name: TICKET\_CATEGORY  
  
Entity Description: This entity in our database provides information about the various categories of tickets available at the concert. It connects to only one table; TICKETS.

Main Attributes of TICKET\_CATEGORY:   
tkt\_category\_id: (Primary Key) A unique ID for each category of the ticket.  
ticket\_id: (Foreign key) attribute to reference back to TICKETS table.   
description: A description of the ticket category which depends on various factors like price.(i.e. VIP tickets, General Admission, other special categories)

price: Stores information about the price of each ticket category.

start\_date: The earliest number of days prior at which a ticket of the category can be purchased.

end\_date: The maximum number of days prior to a show, which a ticket of the category can be purchased.

Entity Name: CUSTOMER  
  
Entity Description: This table contains all the relevant information about all the customers attending concert. It connects to only one table; ORDER\_DETAILS.

Main Attributes of CUSTOMER:  
customer\_id: (Primary Key) A unique ID for each customer who purchased a ticket.  
customer\_name: The customer’s full name.   
customer\_email: The customer’s email address, it is a unique attribute and can be used as a secondary key.

customer\_contact: Stores information about contact details of the customer.

customer\_code: A confirmation code sent to customer while purchasing the ticket.

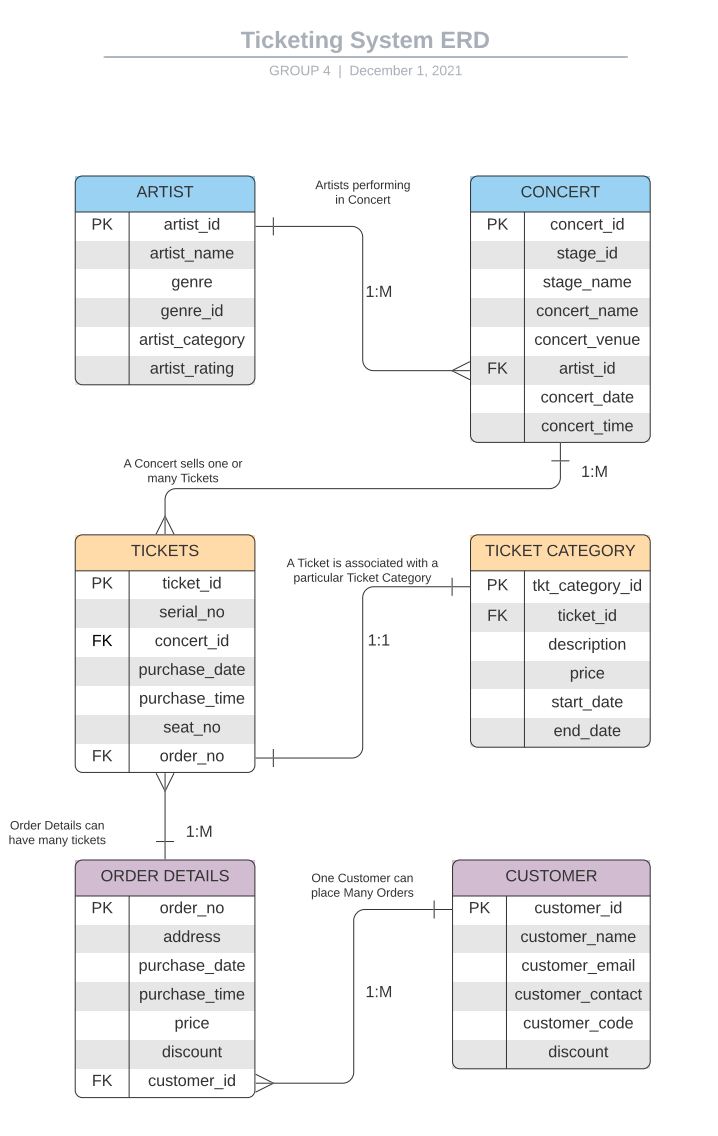
discount: Discount given to customer (if any)

Entity Name: ORDER\_DETAILS  
  
Entity Description: Table that stores relevant information about the purchase details of the customer. It references back to 2 tables: TICKETS and CUSTOMER

Main Attributes of ORDER\_DETAILS:  
order\_no: (Primary Key) A unique ID for each order placed by the customer.  
address: Address at which customer wants the tickets to be delivered to.   
purchase\_date: date at which order is placed by the customer.

purchase\_time: time at which order is placed by the customer.

Discount: Discount given to customer (if any)  
customer\_id: (Foreign key) attribute which references to CUSTOMER table.



**DDL Source Code:**

/\* DROP statements to clean up objects from previous run \*/

-- Triggers

DROP TRIGGER TRG\_ARTIST;

DROP TRIGGER TRG\_CONCERT;

DROP TRIGGER TRG\_TICKETS;

DROP TRIGGER TRG\_TICKET\_CATEGORY;

DROP TRIGGER TRG\_ORDER\_DETAILS;

DROP TRIGGER TRG\_CUSTOMER;

-- Sequences

DROP SEQUENCE SEQ\_CONCERT\_concert\_id;

DROP SEQUENCE SEQ\_TICKETS\_ticket\_id;

DROP SEQUENCE SEQ\_ORDER\_DETAILS\_order\_no;

DROP SEQUENCE SEQ\_CUSTOMER\_customer\_id;

-- Views

DROP VIEW ARTISTInfo;

DROP VIEW CONCERTInfo;

DROP VIEW TICKETSInfo;

DROP VIEW CUSTOMERInfo;

-- Indices

DROP INDEX IDX\_customer\_name ;

DROP INDEX IDX\_order\_no;

DROP INDEX IDX\_Order\_Details\_customer\_id\_FK ;

DROP INDEX IDX\_Tickets\_ticket\_id;

DROP INDEX IDX\_Tickets\_concert\_id\_FK;

DROP INDEX IDX\_Tickets\_tkt\_category\_id\_FK;

DROP INDEX IDX\_Tickets\_serial\_no;

DROP INDEX IDX\_Ticket\_Category\_ticket\_id\_FK;

DROP INDEX IDX\_Ticket\_Category\_price;

DROP INDEX IDX\_Ticket\_Category\_description;

DROP INDEX IDX\_Concert\_concert\_name;

DROP INDEX IDX\_Concert\_concert\_venue;

DROP INDEX IDX\_Concert\_concert\_date;

DROP INDEX IDX\_IDX\_Concert\_artist\_id\_FK ;

DROP INDEX IDX\_Artist\_genre\_id;

DROP INDEX IDX\_Artist\_concert\_id\_FK;

DROP INDEX IDX\_Artist\_Artist\_name;

DROP INDEX IDX\_Artist\_genre;

--Drop Primary Keys

ALTER TABLE ARTIST DROP Constraint PK\_Artist CASCADE;

ALTER TABLE CONCERT DROP Constraint PK\_Concert CASCADE;

ALTER TABLE TICKET\_CATEGORY DROP Constraint PK\_TICKET\_CATEGORY CASCADE;

ALTER TABLE TICKETS DROP Constraint PK\_TICKETS CASCADE;

-- Tables

DROP TABLE ARTIST;

DROP TABLE CONCERT;

DROP TABLE TICKETS;

DROP TABLE TICKET\_CATEGORY;

DROP TABLE ORDER\_DETAILS;

DROP TABLE CUSTOMER;

/\* Create tables based on entities \*/

CREATE TABLE ARTIST (

artist\_id INTEGER NOT NULL,

artist\_name VARCHAR2(30) NOT NULL,

genre VARCHAR2(30) NOT NULL,

genre\_id INTEGER,

artist\_category VARCHAR2(30),

artist\_rating INTEGER,

CONSTRAINT PK\_ARTIST PRIMARY KEY (artist\_id)

);

CREATE TABLE CONCERT (

concert\_id INTEGER NOT NULL,

stage\_id INTEGER NOT NULL,

stage\_name VARCHAR2(128) NOT NULL,

concert\_name VARCHAR2(512) NOT NULL,

concert\_venue VARCHAR2(30) NOT NULL,

artist\_id INTEGER,

concert\_date DATE NOT NULL,

concert\_time VARCHAR2(30) NOT NULL,

CONSTRAINT PK\_CONCERT PRIMARY KEY (concert\_id),

CONSTRAINT FK\_CONCERT\_artist\_id FOREIGN KEY (artist\_id) REFERENCES ARTIST

);

CREATE TABLE CUSTOMER (

customer\_id INTEGER NOT NULL,

customer\_name VARCHAR2(30) NOT NULL,

customer\_email VARCHAR2(30) NOT NULL,

customer\_contact VARCHAR2(30),

customer\_code INTEGER,

discount INTEGER,

CONSTRAINT PK\_CUSTOMER PRIMARY KEY (customer\_id)

);

CREATE TABLE ORDER\_DETAILS (

order\_no INTEGER NOT NULL,

address VARCHAR(30) NOT NULL,

purchase\_date DATE NOT NULL,

purchase\_time VARCHAR(30) NOT NULL,

price INTEGER NOT NULL,

discount INTEGER NOT NULL,

customer\_id INTEGER ,

CONSTRAINT PK\_ORDER\_DETAILS PRIMARY KEY (order\_no),

CONSTRAINT FK\_ORDER\_DETAILS\_customer\_id FOREIGN KEY (customer\_id) REFERENCES CUSTOMER

);

CREATE TABLE TICKETS (

ticket\_id INTEGER NOT NULL,

serial\_no INTEGER,

concert\_id INTEGER NOT NULL,

purchase\_date DATE NOT NULL,

purchase\_time varchar(30),

seat\_no VARCHAR(30) NOT NULL,

order\_no INTEGER,

CONSTRAINT PK\_TICKETS PRIMARY KEY (ticket\_id),

CONSTRAINT FK\_TICKETS\_order\_no FOREIGN KEY (order\_no) REFERENCES ORDER\_DETAILS,

CONSTRAINT FK\_TICKETS\_concert\_id FOREIGN KEY (concert\_id) REFERENCES CONCERT

);

CREATE TABLE TICKET\_CATEGORY (

tkt\_category\_id INTEGER NOT NULL,

ticket\_id INTEGER ,

description VARCHAR2(1024) NOT NULL,

price INTEGER NOT NULL,

start\_date DATE,

end\_date DATE,

CONSTRAINT PK\_TICKET\_CATEGORY PRIMARY KEY (tkt\_category\_id),

CONSTRAINT FK\_TICKET\_CATEGORY\_ticket\_id FOREIGN KEY (ticket\_id) REFERENCES TICKETS

);

/\* Create indices for natural keys, foreign keys, and frequently-queried columns \*/

-- Customer

-- Natural Keys

CREATE INDEX IDX\_customer\_name ON Customer (customer\_name);

-- Order\_Details

-- Foreign Keys

CREATE INDEX IDX\_Order\_Details\_customer\_id\_FK ON Order\_Details (customer\_id);

-- Tickets

-- Frequently-queried columns

CREATE INDEX IDX\_Tickets\_serial\_no ON Tickets (serial\_no);

-- Ticket\_Category

-- Foreign Keys

CREATE INDEX IDX\_Ticket\_Category\_ticket\_id\_FK ON Ticket\_Category (ticket\_id);

-- Frequently-queried columns

CREATE INDEX IDX\_Ticket\_Category\_price ON Ticket\_Category (price);

CREATE INDEX IDX\_Ticket\_Category\_description ON Ticket\_Category (description);

-- Concert

-- Foreign Keys

CREATE INDEX IDX\_Concert\_artist\_id\_FK ON Concert (artist\_id);

-- Frequently-queried columns

CREATE INDEX IDX\_Concert\_concert\_name ON Concert (concert\_name);

CREATE INDEX IDX\_Concert\_concert\_venue ON Concert (concert\_venue);

CREATE INDEX IDX\_Concert\_concert\_date ON Concert (concert\_date);

-- Artist

-- Natural Keys

CREATE INDEX IDX\_Artist\_genre\_id ON Artist (genre\_id);

-- Frequently-queried columns

CREATE INDEX IDX\_Artist\_Artist\_name ON Artist (Artist\_name);

CREATE INDEX IDX\_Artist\_genre ON Artist (genre);

/\* Alter Tables by adding Audit Columns \*/

ALTER TABLE CUSTOMER ADD (

created\_by VARCHAR2(30),

date\_created DATE,

modified\_by VARCHAR2(30),

date\_modified DATE

);

ALTER TABLE ORDER\_DETAILS ADD (

created\_by VARCHAR2(30),

date\_created DATE,

modified\_by VARCHAR2(30),

date\_modified DATE

);

ALTER TABLE TICKETS ADD (

created\_by VARCHAR2(30),

date\_created DATE,

modified\_by VARCHAR2(30),

date\_modified DATE

);

ALTER TABLE TICKET\_CATEGORY ADD (

created\_by VARCHAR2(30),

date\_created DATE,

modified\_by VARCHAR2(30),

date\_modified DATE

);

ALTER TABLE CONCERT ADD (

created\_by VARCHAR2(30),

date\_created DATE,

modified\_by VARCHAR2(30),

date\_modified DATE

);

ALTER TABLE ARTIST ADD (

created\_by VARCHAR2(30),

date\_created DATE,

modified\_by VARCHAR2(30),

date\_modified DATE

);

/\* Create Views \*/

-- Business purpose: The CustomerInfo view will be used primarily for rapidly fetching information about customer for entry to the concert.

CREATE OR REPLACE VIEW CustomerInfo AS

SELECT customer\_id,customer\_name,customer\_code,customer\_email

FROM CUSTOMER;

-- Business purpose: The ArtistInfo view will be used to fetch information about an artist to be displayed as part of detail information of the artist performing.

CREATE OR REPLACE VIEW ArtistInfo AS

SELECT artist\_id,artist\_name,genre,artist\_rating,artist\_category

FROM ARTIST;

-- Business purpose: The ConcertInfo view will be used to populate information about the concert where the concert will take place.

CREATE OR REPLACE VIEW ConcertInfo AS

SELECT concert\_id,concert\_name,concert\_venue,concert\_date,concert\_time

FROM CONCERT;

-- Business purpose: The TicketCat view will be used to populate a list of all the tickets sold for all the concerts.

CREATE OR REPLACE VIEW TicketCat AS

SELECT tkt\_category\_id,description,price,start\_date,end\_date

FROM TICKET\_CATEGORY;

/\* Create Sequences \*/

CREATE SEQUENCE SEQ\_CONCERT\_concert\_id

INCREMENT BY 1

START WITH 0

NOMAXVALUE

MINVALUE 0

NOCACHE;

CREATE SEQUENCE SEQ\_TICKETS\_ticket\_id

INCREMENT BY 1

START WITH 0

NOMAXVALUE

MINVALUE 0

NOCACHE;

CREATE SEQUENCE SEQ\_ORDER\_DETAILS\_order\_no

INCREMENT BY 1

START WITH 0

NOMAXVALUE

MINVALUE 0

NOCACHE;

CREATE SEQUENCE SEQ\_CUSTOMER\_customer\_id

INCREMENT BY 1

START WITH 0

NOMAXVALUE

MINVALUE 0

NOCACHE;

/\* Create Triggers \*/

-- Business purpose: The TRG\_CUSTOMER trigger automatically assigns a sequential customer ID to a newly-inserted row in the CUSTOMER table, as well as assigning appropriate values to the created\_by and date\_created fields. If the record is being inserted or updated, appropriate values are assigned to the modified\_by and modified\_date fields.

CREATE OR REPLACE TRIGGER TRG\_CUSTOMER

BEFORE INSERT OR UPDATE ON CUSTOMER

FOR EACH ROW

BEGIN

IF INSERTING THEN

IF :NEW.customer\_id IS NULL THEN

:NEW.customer\_id := SEQ\_CUSTOMER\_customer\_id.NEXTVAL;

END IF;

IF :NEW.created\_by IS NULL THEN

:NEW.created\_by := USER;

END IF;

IF :NEW.date\_created IS NULL THEN

:NEW.date\_created := SYSDATE;

END IF;

END IF;

IF INSERTING OR UPDATING THEN

:NEW.modified\_by := USER;

:NEW.date\_modified := SYSDATE;

END IF;

END;

/

-- Business purpose: The TRG\_ORDER\_DETAILS trigger automatically assigns a sequential order\_no to a newly-inserted row in the ORDER\_DETAILS table, as well as setting the join date to the current system date and assigning appropriate values to the created\_by and date\_created fields. If the record is being inserted or updated, appropriate values are assigned to the modified\_by and modified\_date fields.

CREATE OR REPLACE TRIGGER TRG\_ORDER\_DETAILS

BEFORE INSERT OR UPDATE ON ORDER\_DETAILS

FOR EACH ROW

BEGIN

IF INSERTING THEN

IF :NEW.order\_no IS NULL THEN

:NEW.order\_no := SEQ\_ORDER\_DETAILS\_order\_no.NEXTVAL;

END IF;

IF :NEW.created\_by IS NULL THEN

:NEW.created\_by := USER;

END IF;

IF :NEW.date\_created IS NULL THEN

:NEW.date\_created := SYSDATE;

END IF;

END IF;

IF INSERTING OR UPDATING THEN

:NEW.modified\_by := USER;

:NEW.date\_modified := SYSDATE;

END IF;

END;

/

-- Business purpose: The TRG\_TICKET\_CATEGORY trigger sets the modified\_by and date\_modified fields to appropriate values in a newly inserted or updated record; if the record is being inserted, then the created\_by and date\_created fields are set to appropriate values too.

CREATE OR REPLACE TRIGGER TRG\_TICKET\_CATEGORY

BEFORE INSERT OR UPDATE ON TICKET\_CATEGORY

FOR EACH ROW

BEGIN

IF INSERTING THEN

IF :NEW.created\_by IS NULL THEN

:NEW.created\_by := USER;

END IF;

IF :NEW.date\_created IS NULL THEN

:NEW.date\_created := SYSDATE;

END IF;

END IF;

IF INSERTING OR UPDATING THEN

:NEW.modified\_by := USER;

:NEW.date\_modified := SYSDATE;

END IF;

END;

/

-- Business purpose: The TRG\_TICKET trigger automatically assigns a sequential ticket\_id to a newly-inserted row in the TICKET table, as well as setting the join date to the current system date and assigning appropriate values to the created\_by and date\_created fields. If the record is being inserted or updated, appropriate values are assigned to the modified\_by and modified\_date fields.

CREATE OR REPLACE TRIGGER TRG\_TICKETS

BEFORE INSERT OR UPDATE ON TICKETS

FOR EACH ROW

BEGIN

IF INSERTING THEN

IF :NEW.ticket\_id IS NULL THEN

:NEW.ticket\_id := SEQ\_TICKETS\_ticket\_id.NEXTVAL;

END IF;

IF :NEW.created\_by IS NULL THEN

:NEW.created\_by := USER;

END IF;

IF :NEW.date\_created IS NULL THEN

:NEW.date\_created := SYSDATE;

END IF;

END IF;

IF INSERTING OR UPDATING THEN

:NEW.modified\_by := USER;

:NEW.date\_modified := SYSDATE;

END IF;

END;

/

-- Business purpose: The TRG\_CONCERT trigger automatically assigns a sequential concert\_id to a newly-inserted row in the CONCERT table, as well as setting the join date to the current system date and assigning appropriate values to the created\_by and date\_created fields. If the record is being inserted or updated, appropriate values are assigned to the modified\_by and modified\_date fields.

CREATE OR REPLACE TRIGGER TRG\_CONCERT

BEFORE INSERT OR UPDATE ON CONCERT

FOR EACH ROW

BEGIN

IF INSERTING THEN

IF :NEW.concert\_id IS NULL THEN

:NEW.concert\_id := SEQ\_CONCERT\_concert\_id.NEXTVAL;

END IF;

IF :NEW.created\_by IS NULL THEN

:NEW.created\_by := USER;

END IF;

IF :NEW.date\_created IS NULL THEN

:NEW.date\_created := SYSDATE;

END IF;

END IF;

IF INSERTING OR UPDATING THEN

:NEW.modified\_by := USER;

:NEW.date\_modified := SYSDATE;

END IF;

END;

/

-- Business purpose: The TRG\_ARTIST trigger sets the modified\_by and date\_modified fields to appropriate values in a newly inserted or updated record; if the record is being inserted, then the created\_by and date\_created fields are set to appropriate values too.

CREATE OR REPLACE TRIGGER TRG\_ARTIST

BEFORE INSERT OR UPDATE ON ARTIST

FOR EACH ROW

BEGIN

IF INSERTING THEN

IF :NEW.created\_by IS NULL THEN

:NEW.created\_by := USER;

END IF;

IF :NEW.date\_created IS NULL THEN

:NEW.date\_created := SYSDATE;

END IF;

END IF;

IF INSERTING OR UPDATING THEN

:NEW.modified\_by := USER;

:NEW.date\_modified := SYSDATE;

END IF;

END;

/

-- Check the DBMS data dictionary to make sure that all objects have been created successfully

SELECT TABLE\_NAME FROM USER\_TABLES;

SELECT OBJECT\_NAME, STATUS, CREATED, LAST\_DDL\_TIME FROM USER\_OBJECTS;

--Artist

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (001,'John Joy','Rock',001,'Young Adult',6);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (007,'Hozier','Pop',002,'Adult',7);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (099,'Lady Yaya','Rap',003,'Explicit',8);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (101,'Harry Patter','Indie',004,'Young Adult',7);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (021,'Kanye South','Jazz',005,'Adult',8);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (030,'Taylor Drift','Rock',001,'Explicit',9);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (028,'Pre Malone','Rap',003,'Adult',5);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (066,'Twenty Two Pilots','EDM',006,'Explicit',9);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (120,'Pitstop','Pop',002,'Adult',7);

INSERT INTO Artist (artist\_id, artist\_name,genre, genre\_id,artist\_category, artist\_rating)

VALUES (089,'The Weekdy','Indie',004,'Young Adult',8);

--Concert

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (001,03,'Green Stage','RockFeast','Iron City','001','25-DEC-2021','19:00:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (001,03,'Green Stage','RockFeast','Iron City','001','25-DEC-2021','19:00:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (020,11,'Red Stage','PopFeast','Wild Mikes','007','12-NOV-2022','20:00:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (021,26,'Blue Stage','RapFeast','Royce Hall','099','12-JAN-2022','19:30:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (030,19,'Red Stage','IndieFeast','Wild Mikes','101','15-DEC-2021','20:00:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (102,07,'Green Stage','JazzFeast','House of Blues','021','12-NOV-2022','19:30:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (111,04,'Green Stage','RockFeast','Petco Park','030','12-FEB-2022','20:00:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (018,99,'Red Stage','RapFeast','Oracle Park','028','12-MAY-2022','21:00:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (091,28,'Yellow Stage','EDMFeast','Iron City','066','12-JAN-2022','18:30:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (071,32,'Blue Stage','PopFeast','House of Blues','120','12-MAY-2022','18:00:00');

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (011,68,'Yellow Stage','IndieFeast','Oracle Park','089','12-JUN-2022','20:00:00');

--Customer

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (231,'Adam','Adam.A@gmail.com','4691231234',03,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (232,'Adam','A324@gmail.com','4691231235',03,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (233,'steve','st342@gmail.com','4691231236',03,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (234,'Polo','Aer34@gmail.com','4691231237',03,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (235,'Alan','Adas34@gmail.com','4691231238',03,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (236,'Jeff','jg324A@gmail.com','4691231239',03,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (237,'steve','stve45@gmail.com','4691231210',03,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (238,'ry','rder34@gmail.com','4691231211',06,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (239,'wanda','wasdf3@gmail.com','4691231212',03,10);

INSERT INTO Customer(customer\_id, customer\_name, customer\_email, customer\_contact, customer\_code, discount)

VALUES (240,'Lovo','43234lop@gmail.com','4691231213',03,10);

Select \* from customer;

--Order Details

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (23,'125 Main Rd','08-Dec-2008','19:00:00',200,10,231);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (24,'126 Main Rd','05-NOv-2010','19:00:00',200,10,232);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (25,'127 Main Rd','01-NOv-2008','19:00:00',200,10,233);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (26,'128 Main Rd','02-Jan-2009','19:00:00',200,10,234);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (27,'129 Main Rd','07-Jan-2011','19:00:00',200,10,235);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (28,'132 Main Rd','11-Apr-2010','19:00:00',200,10,236);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (29,'133 Main Rd','11-Mar-2009','19:00:00',200,10,237);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (30,'134 Main Rd','10-Oct-2008','19:00:00',200,10,238);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (21,'123 Main Rd','11-Nov-2008','19:00:00',200,10,239);

INSERT INTO Order\_Details(order\_no, address, purchase\_date, purchase\_time, price, discount, customer\_id)

VALUES (22,'124 Main Rd','06-Jan-2009','19:00:00',200,10,240);

--Tickets

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (111,1101,001,'11-Nov-2008','19:00:00','A23',21);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (112,1102,020,'06-Jan-2009','19:00:00','A24',22);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (113,1103,021,'08-Dec-2008','19:00:00','A25',23);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (114,1104,030,'05-NOv-2010','19:00:00','A26',24);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (115,1105,102,'01-NOv-2008','19:00:00','A27',25);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (116,1106,111,'02-Jan-2009','19:00:00','A28',26);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (117,1107,018,'07-Jan-2011','19:00:00','A29',27);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (118,1108,091,'11-Apr-2010','19:00:00','A30',28);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (119,1109,071,'11-Mar-2009','19:00:00','A31',29);

INSERT INTO Tickets (ticket\_id, serial\_no, concert\_id, purchase\_date, purchase\_time, seat\_no, order\_no )

VALUES (120,1110,011,'10-Oct-2008','19:00:00','A32',30);

--Ticket Category

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (011,111,'Front Left Section',200,'01-Nov-2012','02-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (022,112,'Front Right Section',200,'03-Nov-2012','04-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (033,113,'Back Left Section',200,'05-Nov-2012','06-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (044,114,'Back Right Section',200,'07-Nov-2012','08-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (055,115,'Balcony Front Left Section',200,'09-Nov-2012','10-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (066,116,'Balcony Front Right Section',200,'11-Nov-2012','12-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (077,117,'Balcony Back Left Section',200,'13-Nov-2012','14-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (088,118,'Balcony Back Right Section',200,'15-Nov-2012','16-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (099,119,'SideBar Right',200,'17-Nov-2012','18-Nov-2012');

INSERT INTO Ticket\_Category(tkt\_category\_id, ticket\_id, description, price, start\_date, end\_date)

VALUES (1000,120,'SideBar Left',200,'19-Nov-2012','20-Nov-2012');

select \* from Ticket\_Category;

--Q1) Select all columns and all rows from one table

SELECT \* FROM artist;

--Q2) Select five columns and all rows from one table

SELECT artist\_id, artist\_name, genre, genre\_id, artist\_category

FROM artist;

--Q3) Select all columns from all rows from one view

SELECT \* FROM CustomerInfo;

--Q4)Using a join on 2 tables, select all columns and all rows

SELECT \* FROM concert LEFT OUTER JOIN Tickets ON concert.concert\_id = tickets.concert\_id;

--Q5) Select and order data retrieved from one table

SELECT \* FROM Artist

ORDER BY artist\_id;

--Q6) Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 10 rows

SELECT artist.artist\_id, concert.concert\_name, concert.stage\_id, concert.concert\_venue, tickets.ticket\_id

FROM artist INNER JOIN concert ON artist.artist\_id = concert.artist\_id

INNER JOIN tickets ON concert.concert\_id = tickets.concert\_id

FETCH FIRST 10 ROWS ONLY;

--Q7) Select distinct rows using joins on 3 tables

SELECT DISTINCT \*

FROM concert INNER JOIN tickets ON concert.concert\_id = tickets.concert\_id

INNER JOIN artist ON concert.artist\_id = artist.artist\_id;

--Q8)Use GROUP BY and HAVING in a select statement using one or more tables

SELECT concert.concert\_name, AVG(artist.artist\_rating)

FROM concert INNER JOIN artist ON concert.artist\_id = artist.artist\_id

GROUP BY concert.concert\_name, concert.concert\_id

HAVING concert.concert\_id = 021;

--Q9) Use IN clause to select data from one or more tables

SELECT \* FROM concert

WHERE concert\_id IN (020, 021, 030);

-- Q10. Select length of one column from one table (use LENGTH function)

SELECT LENGTH(customer\_id) FROM CUSTOMER;

-- Q11. Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement. Make sure you use ROLLBACK afterward so that the data will not be physically removed

SELECT \* FROM CUSTOMER;

DELETE FROM CUSTOMER

WHERE customer\_id = 231;

SELECT \* FROM CUSTOMER;

ROLLBACK;

-- Q12. Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement. Make sure you use ROLLBACK afterward so that the data will not be physically removed

SELECT \* FROM ARTIST;

UPDATE ARTIST

SET artist\_name= 'The Weeknd'

WHERE artist\_id = 101;

SELECT \* FROM ARTIST;

ROLLBACK;

-- Q13. List all artist names, their categories, and their ratings, ordered alphabetically by their name where the genre is pop.

SELECT a.artist\_name, a.artist\_category, a.artist\_rating

FROM ARTIST a

WHERE a.genre = 'pop'

ORDER BY a.artist\_name;

-- Q14. List all unique ticket categories; display the average price along where the description is 'fun'.

SELECT AVG(price)AS avg\_price

FROM TICKET\_CATEGORY

WHERE description = 'fun';

-- Q15. Display the artist\_id and artist\_name of all artists who will perform more than once.

SELECT a.artist\_id, a.artist\_name

FROM ARTIST a GROUP BY a.artist\_id, a.artist\_name

HAVING COUNT(a.artist\_id) > 1;

-- Q16. Top 3 artists basis on the no of concerts performed.

select a.artist\_name,count(c.concert\_id) as total\_no\_concerts from artist a

join concert c on a.artist\_id = c.artist\_id

group by a.artist\_name

order by total\_no\_concerts

fetch first 3 rows only

-- Q17.Top 3 prices of tickets sold

SELECT Price from order\_details

Where price > 23;

-- Q18. Top 3 customers which has the highest no of tickets purchased and how much did they spend in total

select count(order\_no),customer\_id,sum(price) from order\_details

group by customer\_id

order by count(order\_no)

fetch first 3 rows only

-- Q19. Top 35 customers which has the highest no of tickets purchased and how much did they spend in total

select count(order\_no),customer\_id,sum(price) from order\_details

group by customer\_id

order by count(order\_no)

fetch first 35 rows only ;

-- Q20. Top 15 customers which has the highest no of tickets purchased and how much did they spend in total

select count(order\_no),customer\_id,sum(price) from order\_details

group by customer\_id

order by count(order\_no)

fetch first 15 rows only;

**SQL Outputs:**

Trigger TRG\_ARTIST dropped.

Trigger TRG\_CONCERT dropped.

Trigger TRG\_TICKETS dropped.

Trigger TRG\_TICKET\_CATEGORY dropped.

Trigger TRG\_ORDER\_DETAILS dropped.

Trigger TRG\_CUSTOMER dropped.

Sequence SEQ\_CONCERT\_CONCERT\_ID dropped.

Sequence SEQ\_TICKETS\_TICKET\_ID dropped.

Sequence SEQ\_ORDER\_DETAILS\_ORDER\_NO dropped.

Sequence SEQ\_CUSTOMER\_CUSTOMER\_ID dropped.

View ARTISTINFO dropped.

View CONCERTINFO dropped.

Error starting at line : 21 in command -

DROP VIEW TICKETSInfo

Error report -

ORA-00942: table or view does not exist

00942. 00000 - "table or view does not exist"

\*Cause:

\*Action:

View CUSTOMERINFO dropped.

Index IDX\_CUSTOMER\_NAME dropped.

Error starting at line : 27 in command -

DROP INDEX IDX\_order\_no

Error report -

ORA-01418: specified index does not exist

01418. 00000 - "specified index does not exist"

\*Cause:

\*Action:

Index IDX\_ORDER\_DETAILS\_CUSTOMER\_ID\_FK dropped.

Error starting at line : 30 in command -

DROP INDEX IDX\_Tickets\_ticket\_id

Error report -

ORA-01418: specified index does not exist

01418. 00000 - "specified index does not exist"

\*Cause:

\*Action:

Error starting at line : 31 in command -

DROP INDEX IDX\_Tickets\_concert\_id\_FK

Error report -

ORA-01418: specified index does not exist

01418. 00000 - "specified index does not exist"

\*Cause:

\*Action:

Error starting at line : 32 in command -

DROP INDEX IDX\_Tickets\_tkt\_category\_id\_FK

Error report -

ORA-01418: specified index does not exist

01418. 00000 - "specified index does not exist"

\*Cause:

\*Action:

Index IDX\_TICKETS\_SERIAL\_NO dropped.

Index IDX\_TICKET\_CATEGORY\_TICKET\_ID\_FK dropped.

Index IDX\_TICKET\_CATEGORY\_PRICE dropped.

Index IDX\_TICKET\_CATEGORY\_DESCRIPTION dropped.

Index IDX\_CONCERT\_CONCERT\_NAME dropped.

Index IDX\_CONCERT\_CONCERT\_VENUE dropped.

Index IDX\_CONCERT\_CONCERT\_DATE dropped.

Error starting at line : 42 in command -

DROP INDEX IDX\_IDX\_Concert\_artist\_id\_FK

Error report -

ORA-01418: specified index does not exist

01418. 00000 - "specified index does not exist"

\*Cause:

\*Action:

Index IDX\_ARTIST\_GENRE\_ID dropped.

Error starting at line : 45 in command -

DROP INDEX IDX\_Artist\_concert\_id\_FK

Error report -

ORA-01418: specified index does not exist

01418. 00000 - "specified index does not exist"

\*Cause:

\*Action:

Index IDX\_ARTIST\_ARTIST\_NAME dropped.

Index IDX\_ARTIST\_GENRE dropped.

Table ARTIST altered.

Table CONCERT altered.

Table TICKET\_CATEGORY altered.

Table TICKETS altered.

Table ARTIST dropped.

Table CONCERT dropped.

Table TICKETS dropped.

Table TICKET\_CATEGORY dropped.

Table ORDER\_DETAILS dropped.

Table CUSTOMER dropped.

Table ARTIST created.

Table CONCERT created.

Table CUSTOMER created.

Table ORDER\_DETAILS created.

Table TICKETS created.

Table TICKET\_CATEGORY created.

Index IDX\_CUSTOMER\_NAME created.

Index IDX\_ORDER\_DETAILS\_CUSTOMER\_ID\_FK created.

Index IDX\_TICKETS\_SERIAL\_NO created.

Index IDX\_TICKET\_CATEGORY\_TICKET\_ID\_FK created.

Index IDX\_TICKET\_CATEGORY\_PRICE created.

Index IDX\_TICKET\_CATEGORY\_DESCRIPTION created.

Index IDX\_CONCERT\_ARTIST\_ID\_FK created.

Index IDX\_CONCERT\_CONCERT\_NAME created.

Index IDX\_CONCERT\_CONCERT\_VENUE created.

Index IDX\_CONCERT\_CONCERT\_DATE created.

Index IDX\_ARTIST\_GENRE\_ID created.

Index IDX\_ARTIST\_ARTIST\_NAME created.

Index IDX\_ARTIST\_GENRE created.

Table CUSTOMER altered.

Table ORDER\_DETAILS altered.

Table TICKETS altered.

Table TICKET\_CATEGORY altered.

Table CONCERT altered.

Table ARTIST altered.

View CUSTOMERINFO created.

View ARTISTINFO created.

View CONCERTINFO created.

View TICKETCAT created.

Sequence SEQ\_CONCERT\_CONCERT\_ID created.

Sequence SEQ\_TICKETS\_TICKET\_ID created.

Sequence SEQ\_ORDER\_DETAILS\_ORDER\_NO created.

Sequence SEQ\_CUSTOMER\_CUSTOMER\_ID created.

Trigger TRG\_CUSTOMER compiled

Trigger TRG\_ORDER\_DETAILS compiled

Trigger TRG\_TICKET\_CATEGORY compiled

Trigger TRG\_TICKETS compiled

Trigger TRG\_CONCERT compiled

Trigger TRG\_ARTIST compiled

TABLE\_NAME

--------------------------------------------------------------------------------------------------------------------------------

CONCERT

ARTIST

CUSTOMER

ORDER\_DETAILS

TICKETS

TICKET\_CATEGORY

6 rows selected.

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

CUSTOMERINFO VALID 01-DEC-21 01-DEC-21

ARTISTINFO VALID 01-DEC-21 01-DEC-21

CONCERTINFO VALID 01-DEC-21 01-DEC-21

TICKETCAT VALID 27-NOV-21 01-DEC-21

TRG\_CUSTOMER VALID 01-DEC-21 01-DEC-21

TRG\_ORDER\_DETAILS VALID 01-DEC-21 01-DEC-21

TRG\_TICKET\_CATEGORY VALID 01-DEC-21 01-DEC-21

TRG\_TICKETS VALID 01-DEC-21 01-DEC-21

TRG\_CONCERT VALID 01-DEC-21 01-DEC-21

TRG\_ARTIST VALID 01-DEC-21 01-DEC-21

PK\_ARTIST VALID 01-DEC-21 01-DEC-21

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

CONCERT VALID 01-DEC-21 01-DEC-21

ARTIST VALID 01-DEC-21 01-DEC-21

PK\_CONCERT VALID 01-DEC-21 01-DEC-21

CUSTOMER VALID 01-DEC-21 01-DEC-21

PK\_CUSTOMER VALID 01-DEC-21 01-DEC-21

ORDER\_DETAILS VALID 01-DEC-21 01-DEC-21

PK\_ORDER\_DETAILS VALID 01-DEC-21 01-DEC-21

TICKETS VALID 01-DEC-21 01-DEC-21

PK\_TICKETS VALID 01-DEC-21 01-DEC-21

TICKET\_CATEGORY VALID 01-DEC-21 01-DEC-21

PK\_TICKET\_CATEGORY VALID 01-DEC-21 01-DEC-21

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

IDX\_CUSTOMER\_NAME VALID 01-DEC-21 01-DEC-21

IDX\_ORDER\_DETAILS\_CUSTOMER\_ID\_FK VALID 01-DEC-21 01-DEC-21

IDX\_TICKETS\_SERIAL\_NO VALID 01-DEC-21 01-DEC-21

IDX\_TICKET\_CATEGORY\_TICKET\_ID\_FK VALID 01-DEC-21 01-DEC-21

IDX\_TICKET\_CATEGORY\_PRICE VALID 01-DEC-21 01-DEC-21

IDX\_TICKET\_CATEGORY\_DESCRIPTION VALID 01-DEC-21 01-DEC-21

IDX\_CONCERT\_ARTIST\_ID\_FK VALID 01-DEC-21 01-DEC-21

IDX\_CONCERT\_CONCERT\_NAME VALID 01-DEC-21 01-DEC-21

IDX\_CONCERT\_CONCERT\_VENUE VALID 01-DEC-21 01-DEC-21

IDX\_CONCERT\_CONCERT\_DATE VALID 01-DEC-21 01-DEC-21

IDX\_ARTIST\_GENRE\_ID VALID 01-DEC-21 01-DEC-21

OBJECT\_NAME STATUS CREATED LAST\_DDL\_

-------------------------------------------------------------------------------------------------------------------------------- ------- --------- ---------

IDX\_ARTIST\_ARTIST\_NAME VALID 01-DEC-21 01-DEC-21

IDX\_ARTIST\_GENRE VALID 01-DEC-21 01-DEC-21

SEQ\_CONCERT\_CONCERT\_ID VALID 01-DEC-21 01-DEC-21

SEQ\_TICKETS\_TICKET\_ID VALID 01-DEC-21 01-DEC-21

SEQ\_ORDER\_DETAILS\_ORDER\_NO VALID 01-DEC-21 01-DEC-21

SEQ\_CUSTOMER\_CUSTOMER\_ID VALID 01-DEC-21 01-DEC-21

39 rows selected.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

Error starting at line : 453 in command -

INSERT INTO Concert (concert\_id, stage\_id, stage\_name, concert\_name, concert\_venue,artist\_id,concert\_date, concert\_time)

VALUES (001,03,'Green Stage','RockFeast','Iron City','001','25-DEC-2021','19:00:00')

Error report -

ORA-00001: unique constraint (TESTDB.PK\_CONCERT) violated

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

CUSTOMER\_ID CUSTOMER\_NAME CUSTOMER\_EMAIL CUSTOMER\_CONTACT CUSTOMER\_CODE DISCOUNT CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

----------- ------------------------------ ------------------------------ ------------------------------ ------------- ---------- ------------------------------ --------- ------------------------------ ---------

231 Adam Adam.A@gmail.com 4691231234 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

232 Adam A324@gmail.com 4691231235 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

233 steve st342@gmail.com 4691231236 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

234 Polo Aer34@gmail.com 4691231237 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

235 Alan Adas34@gmail.com 4691231238 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

236 Jeff jg324A@gmail.com 4691231239 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

237 steve stve45@gmail.com 4691231210 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

238 ry rder34@gmail.com 4691231211 6 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

239 wanda wasdf3@gmail.com 4691231212 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

240 Lovo 43234lop@gmail.com 4691231213 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

10 rows selected.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

TKT\_CATEGORY\_ID TICKET\_ID DESCRIPTION PRICE START\_DAT END\_DATE CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

--------------- ---------- ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- ---------- --------- --------- ------------------------------ --------- ------------------------------ ---------

11 111 Front Left Section 200 01-NOV-12 02-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

22 112 Front Right Section 200 03-NOV-12 04-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

33 113 Back Left Section 200 05-NOV-12 06-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

44 114 Back Right Section 200 07-NOV-12 08-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

55 115 Balcony Front Left Section 200 09-NOV-12 10-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

66 116 Balcony Front Right Section 200 11-NOV-12 12-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

77 117 Balcony Back Left Section 200 13-NOV-12 14-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

88 118 Balcony Back Right Section 200 15-NOV-12 16-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

99 119 SideBar Right 200 17-NOV-12 18-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

1000 120 SideBar Left 200 19-NOV-12 20-NOV-12 TESTDB 01-DEC-21 TESTDB 01-DEC-21

10 rows selected.

ARTIST\_ID ARTIST\_NAME GENRE GENRE\_ID ARTIST\_CATEGORY ARTIST\_RATING CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

---------- ------------------------------ ------------------------------ ---------- ------------------------------ ------------- ------------------------------ --------- ------------------------------ ---------

1 John Joy Rock 1 Young Adult 6 TESTDB 01-DEC-21 TESTDB 01-DEC-21

7 Hozier Pop 2 Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

99 Lady Yaya Rap 3 Explicit 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

101 Harry Patter Indie 4 Young Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

21 Kanye South Jazz 5 Adult 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

30 Taylor Drift Rock 1 Explicit 9 TESTDB 01-DEC-21 TESTDB 01-DEC-21

28 Pre Malone Rap 3 Adult 5 TESTDB 01-DEC-21 TESTDB 01-DEC-21

66 Twenty Two Pilots EDM 6 Explicit 9 TESTDB 01-DEC-21 TESTDB 01-DEC-21

120 Pitstop Pop 2 Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

89 The Weekdy Indie 4 Young Adult 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

10 rows selected.

ARTIST\_ID ARTIST\_NAME GENRE GENRE\_ID ARTIST\_CATEGORY

---------- ------------------------------ ------------------------------ ---------- ------------------------------

1 John Joy Rock 1 Young Adult

7 Hozier Pop 2 Adult

99 Lady Yaya Rap 3 Explicit

101 Harry Patter Indie 4 Young Adult

21 Kanye South Jazz 5 Adult

30 Taylor Drift Rock 1 Explicit

28 Pre Malone Rap 3 Adult

66 Twenty Two Pilots EDM 6 Explicit

120 Pitstop Pop 2 Adult

89 The Weekdy Indie 4 Young Adult

10 rows selected.

CUSTOMER\_ID CUSTOMER\_NAME CUSTOMER\_CODE CUSTOMER\_EMAIL

----------- ------------------------------ ------------- ------------------------------

231 Adam 3 Adam.A@gmail.com

232 Adam 3 A324@gmail.com

233 steve 3 st342@gmail.com

234 Polo 3 Aer34@gmail.com

235 Alan 3 Adas34@gmail.com

236 Jeff 3 jg324A@gmail.com

237 steve 3 stve45@gmail.com

238 ry 6 rder34@gmail.com

239 wanda 3 wasdf3@gmail.com

240 Lovo 3 43234lop@gmail.com

10 rows selected.

CONCERT\_ID STAGE\_ID STAGE\_NAME CONCERT\_NAME CONCERT\_VENUE ARTIST\_ID CONCERT\_D CONCERT\_TIME CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI TICKET\_ID SERIAL\_NO CONCERT\_ID PURCHASE\_ PURCHASE\_TIME SEAT\_NO ORDER\_NO CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

---------- ---------- -------------------------------------------------------------------------------------------------------------------------------- -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- ------------------------------ ---------- --------- ------------------------------ ------------------------------ --------- ------------------------------ --------- ---------- ---------- ---------- --------- ------------------------------ ------------------------------ ---------- ------------------------------ --------- ------------------------------ ---------

1 3 Green Stage RockFeast Iron City 1 25-DEC-21 19:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 111 1101 1 11-NOV-08 19:00:00 A23 21 TESTDB 01-DEC-21 TESTDB 01-DEC-21

20 11 Red Stage PopFeast Wild Mikes 7 12-NOV-22 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 112 1102 20 06-JAN-09 19:00:00 A24 22 TESTDB 01-DEC-21 TESTDB 01-DEC-21

21 26 Blue Stage RapFeast Royce Hall 99 12-JAN-22 19:30:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 113 1103 21 08-DEC-08 19:00:00 A25 23 TESTDB 01-DEC-21 TESTDB 01-DEC-21

30 19 Red Stage IndieFeast Wild Mikes 101 15-DEC-21 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 114 1104 30 05-NOV-10 19:00:00 A26 24 TESTDB 01-DEC-21 TESTDB 01-DEC-21

102 7 Green Stage JazzFeast House of Blues 21 12-NOV-22 19:30:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 115 1105 102 01-NOV-08 19:00:00 A27 25 TESTDB 01-DEC-21 TESTDB 01-DEC-21

111 4 Green Stage RockFeast Petco Park 30 12-FEB-22 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 116 1106 111 02-JAN-09 19:00:00 A28 26 TESTDB 01-DEC-21 TESTDB 01-DEC-21

18 99 Red Stage RapFeast Oracle Park 28 12-MAY-22 21:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 117 1107 18 07-JAN-11 19:00:00 A29 27 TESTDB 01-DEC-21 TESTDB 01-DEC-21

91 28 Yellow Stage EDMFeast Iron City 66 12-JAN-22 18:30:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 118 1108 91 11-APR-10 19:00:00 A30 28 TESTDB 01-DEC-21 TESTDB 01-DEC-21

71 32 Blue Stage PopFeast House of Blues 120 12-MAY-22 18:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 119 1109 71 11-MAR-09 19:00:00 A31 29 TESTDB 01-DEC-21 TESTDB 01-DEC-21

11 68 Yellow Stage IndieFeast Oracle Park 89 12-JUN-22 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 120 1110 11 10-OCT-08 19:00:00 A32 30 TESTDB 01-DEC-21 TESTDB 01-DEC-21

10 rows selected.

ARTIST\_ID ARTIST\_NAME GENRE GENRE\_ID ARTIST\_CATEGORY ARTIST\_RATING CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

---------- ------------------------------ ------------------------------ ---------- ------------------------------ ------------- ------------------------------ --------- ------------------------------ ---------

1 John Joy Rock 1 Young Adult 6 TESTDB 01-DEC-21 TESTDB 01-DEC-21

7 Hozier Pop 2 Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

21 Kanye South Jazz 5 Adult 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

28 Pre Malone Rap 3 Adult 5 TESTDB 01-DEC-21 TESTDB 01-DEC-21

30 Taylor Drift Rock 1 Explicit 9 TESTDB 01-DEC-21 TESTDB 01-DEC-21

66 Twenty Two Pilots EDM 6 Explicit 9 TESTDB 01-DEC-21 TESTDB 01-DEC-21

89 The Weekdy Indie 4 Young Adult 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

99 Lady Yaya Rap 3 Explicit 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

101 Harry Patter Indie 4 Young Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

120 Pitstop Pop 2 Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

10 rows selected.

ARTIST\_ID CONCERT\_NAME STAGE\_ID CONCERT\_VENUE TICKET\_ID

---------- -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- ---------- ------------------------------ ----------

1 RockFeast 3 Iron City 111

7 PopFeast 11 Wild Mikes 112

99 RapFeast 26 Royce Hall 113

101 IndieFeast 19 Wild Mikes 114

21 JazzFeast 7 House of Blues 115

30 RockFeast 4 Petco Park 116

28 RapFeast 99 Oracle Park 117

66 EDMFeast 28 Iron City 118

120 PopFeast 32 House of Blues 119

89 IndieFeast 68 Oracle Park 120

10 rows selected.

CONCERT\_ID STAGE\_ID STAGE\_NAME CONCERT\_NAME CONCERT\_VENUE ARTIST\_ID CONCERT\_D CONCERT\_TIME CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI TICKET\_ID SERIAL\_NO CONCERT\_ID PURCHASE\_ PURCHASE\_TIME SEAT\_NO ORDER\_NO CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI ARTIST\_ID ARTIST\_NAME GENRE GENRE\_ID ARTIST\_CATEGORY ARTIST\_RATING CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

---------- ---------- -------------------------------------------------------------------------------------------------------------------------------- -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- ------------------------------ ---------- --------- ------------------------------ ------------------------------ --------- ------------------------------ --------- ---------- ---------- ---------- --------- ------------------------------ ------------------------------ ---------- ------------------------------ --------- ------------------------------ --------- ---------- ------------------------------ ------------------------------ ---------- ------------------------------ ------------- ------------------------------ --------- ------------------------------ ---------

1 3 Green Stage RockFeast Iron City 1 25-DEC-21 19:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 111 1101 1 11-NOV-08 19:00:00 A23 21 TESTDB 01-DEC-21 TESTDB 01-DEC-21 1 John Joy Rock 1 Young Adult 6 TESTDB 01-DEC-21 TESTDB 01-DEC-21

20 11 Red Stage PopFeast Wild Mikes 7 12-NOV-22 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 112 1102 20 06-JAN-09 19:00:00 A24 22 TESTDB 01-DEC-21 TESTDB 01-DEC-21 7 Hozier Pop 2 Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

21 26 Blue Stage RapFeast Royce Hall 99 12-JAN-22 19:30:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 113 1103 21 08-DEC-08 19:00:00 A25 23 TESTDB 01-DEC-21 TESTDB 01-DEC-21 99 Lady Yaya Rap 3 Explicit 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

30 19 Red Stage IndieFeast Wild Mikes 101 15-DEC-21 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 114 1104 30 05-NOV-10 19:00:00 A26 24 TESTDB 01-DEC-21 TESTDB 01-DEC-21 101 Harry Patter Indie 4 Young Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

102 7 Green Stage JazzFeast House of Blues 21 12-NOV-22 19:30:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 115 1105 102 01-NOV-08 19:00:00 A27 25 TESTDB 01-DEC-21 TESTDB 01-DEC-21 21 Kanye South Jazz 5 Adult 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

111 4 Green Stage RockFeast Petco Park 30 12-FEB-22 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 116 1106 111 02-JAN-09 19:00:00 A28 26 TESTDB 01-DEC-21 TESTDB 01-DEC-21 30 Taylor Drift Rock 1 Explicit 9 TESTDB 01-DEC-21 TESTDB 01-DEC-21

18 99 Red Stage RapFeast Oracle Park 28 12-MAY-22 21:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 117 1107 18 07-JAN-11 19:00:00 A29 27 TESTDB 01-DEC-21 TESTDB 01-DEC-21 28 Pre Malone Rap 3 Adult 5 TESTDB 01-DEC-21 TESTDB 01-DEC-21

91 28 Yellow Stage EDMFeast Iron City 66 12-JAN-22 18:30:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 118 1108 91 11-APR-10 19:00:00 A30 28 TESTDB 01-DEC-21 TESTDB 01-DEC-21 66 Twenty Two Pilots EDM 6 Explicit 9 TESTDB 01-DEC-21 TESTDB 01-DEC-21

71 32 Blue Stage PopFeast House of Blues 120 12-MAY-22 18:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 119 1109 71 11-MAR-09 19:00:00 A31 29 TESTDB 01-DEC-21 TESTDB 01-DEC-21 120 Pitstop Pop 2 Adult 7 TESTDB 01-DEC-21 TESTDB 01-DEC-21

11 68 Yellow Stage IndieFeast Oracle Park 89 12-JUN-22 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21 120 1110 11 10-OCT-08 19:00:00 A32 30 TESTDB 01-DEC-21 TESTDB 01-DEC-21 89 The Weekdy Indie 4 Young Adult 8 TESTDB 01-DEC-21 TESTDB 01-DEC-21

10 rows selected.

CONCERT\_NAME AVG(ARTIST.ARTIST\_RATING)

-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- -------------------------

RapFeast 8

CONCERT\_ID STAGE\_ID STAGE\_NAME CONCERT\_NAME CONCERT\_VENUE ARTIST\_ID CONCERT\_D CONCERT\_TIME CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

---------- ---------- -------------------------------------------------------------------------------------------------------------------------------- -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------- ------------------------------ ---------- --------- ------------------------------ ------------------------------ --------- ------------------------------ ---------

20 11 Red Stage PopFeast Wild Mikes 7 12-NOV-22 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21

21 26 Blue Stage RapFeast Royce Hall 99 12-JAN-22 19:30:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21

30 19 Red Stage IndieFeast Wild Mikes 101 15-DEC-21 20:00:00 TESTDB 01-DEC-21 TESTDB 01-DEC-21

LENGTH(CUSTOMER\_ID)

-------------------

3

3

3

3

3

3

3

3

3

3

10 rows selected.

CUSTOMER\_ID CUSTOMER\_NAME CUSTOMER\_EMAIL CUSTOMER\_CONTACT CUSTOMER\_CODE DISCOUNT CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

----------- ------------------------------ ------------------------------ ------------------------------ ------------- ---------- ------------------------------ --------- ------------------------------ ---------

231 Adam Adam.A@gmail.com 4691231234 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

232 Adam A324@gmail.com 4691231235 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

233 steve st342@gmail.com 4691231236 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

234 Polo Aer34@gmail.com 4691231237 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

235 Alan Adas34@gmail.com 4691231238 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

236 Jeff jg324A@gmail.com 4691231239 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

237 steve stve45@gmail.com 4691231210 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

238 ry rder34@gmail.com 4691231211 6 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

239 wanda wasdf3@gmail.com 4691231212 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

240 Lovo 43234lop@gmail.com 4691231213 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

10 rows selected.

Error starting at line : 645 in command -

DELETE FROM CUSTOMER

WHERE customer\_id = 231

Error report -

ORA-02292: integrity constraint (TESTDB.FK\_ORDER\_DETAILS\_CUSTOMER\_ID) violated - child record found

CUSTOMER\_ID CUSTOMER\_NAME CUSTOMER\_EMAIL CUSTOMER\_CONTACT CUSTOMER\_CODE DISCOUNT CREATED\_BY DATE\_CREA MODIFIED\_BY DATE\_MODI

----------- ------------------------------ ------------------------------ ------------------------------ ------------- ---------- ------------------------------ --------- ------------------------------ ---------

231 Adam Adam.A@gmail.com 4691231234 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

232 Adam A324@gmail.com 4691231235 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

233 steve st342@gmail.com 4691231236 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

234 Polo Aer34@gmail.com 4691231237 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

235 Alan Adas34@gmail.com 4691231238 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

236 Jeff jg324A@gmail.com 4691231239 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

237 steve stve45@gmail.com 4691231210 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

238 ry rder34@gmail.com 4691231211 6 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

239 wanda wasdf3@gmail.com 4691231212 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

240 Lovo 43234lop@gmail.com 4691231213 3 10 TESTDB 01-DEC-21 TESTDB 01-DEC-21

10 rows selected.

Rollback complete.

no rows selected

0 rows updated.

no rows selected

Rollback complete.

no rows selected

AVG\_PRICE

----------

no rows selected

Error starting at line : 689 in command -

select a.artist\_name,count(c.concert\_id) as total\_no\_concerts from artist a

join concert c on a.artist\_id = c.artist\_id

group by a.artist\_name

order by total\_no\_concerts

fetch first 3 rows only

-- Q17.Top 3 prices of tickets sold

SELECT Price from order\_details

Where price > 23

Error at Command Line : 700 Column : 1

Error report -

SQL Error: ORA-00933: SQL command not properly ended

00933. 00000 - "SQL command not properly ended"

\*Cause:

\*Action:

Error starting at line : 709 in command -

select count(order\_no),customer\_id,sum(price) from order\_details

group by customer\_id

order by count(order\_no)

fetch first 3 rows only

-- Q19. Top 35 customers which has the highest no of tickets purchased and how much did they spend in total

select count(order\_no),customer\_id,sum(price) from order\_details

group by customer\_id

order by count(order\_no)

fetch first 35 rows only

Error at Command Line : 721 Column : 1

Error report -

SQL Error: ORA-00933: SQL command not properly ended

00933. 00000 - "SQL command not properly ended"

\*Cause:

\*Action:

no rows selected