CONESTOGA
Connect Life and Learning

Database Design and Maintenance

Sample Mock-up

Online Music Store Database Design

Chan Lim

June 8, 2021

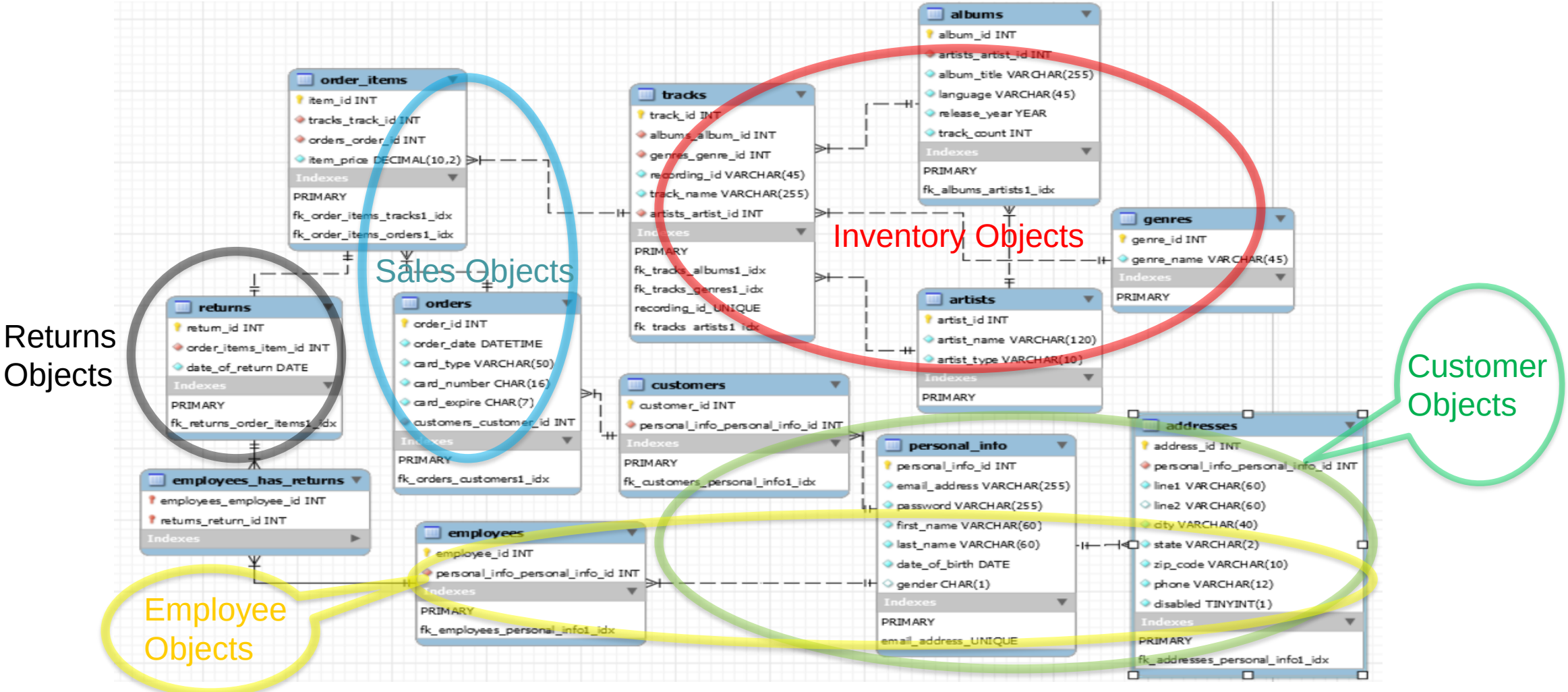
Background

- Our client, Conestoga Music Store (CMS) has a plan to extend their business to online streaming service.
- With so much music available to their customers today, it becomes increasingly challenging to deliver meaningful playlists and recommendations to keep them engaged.
- Normalized data help music services deliver stickier playlists, surfacing a wider range of artists and styles, to help customers find more of the music they love.
- For this, CMS is building the web site for a pilot program and wants to design a database with some sample data to test and demonstrate it for customer management and affiliated business.
- With deep, clean data and standardized artist and recording IDs, CMS will provide an affiliated service to their customers.

Questions and Requirements

1. Grasp the trend of music published in the past three years
   - Genre
   - The number of track recently published for three years
2. What are the songs returned this month and which employees have taken the tasks?
3. Get customers' overall status of order with their background information
   - Music selected and genre
   - Age / Gender
   - Area where they live

Modeling with E / R



Inventory Objects

❖ Artists

Field	Data Type	Constraint	Description	Values
artistID	INT	Primary Key	The unique identifier for the artist.	1
artistName	VARCHAR		The name of the artist.	Aqua
artistType	VARCHAR		The description for the artist type: Male, Female, Mixed Duo, Male Group etc.	Mixed Duo

❖ Albums

Field	Data Type	Constraint	Description	Values
albumID	INT	Primary Key	The unique identifier for the album.	3
artistID	INT	Foreign Key	The identifier for the artist.	9
albumTitle	VARCHAR		The title of the album.	Sippie
language	VARCHAR		The language album was published in.	Japanese
releaseYear	YEAR		When the album was released in.	2016
trackCount	INT		The number of count included in the album.	10

❖ Tracks

Field	Data Type	Constraint	Description	Values
trackID	INT	Primary Key	The unique identifier for the track.	5
artistID	INT	Foreign Key	The identifier for the artist.	4
albumID	INT	Foreign Key	The identifier for the album.	6
genreID	INT	Foreign Key	The identifier for the genre.	3
trackName	VARCHAR		The name of the track.	Hard Times
recordingID	VARCHAR	Unique Key	The unique identifier for the recording.	182914703

❖ Genres

Field	Data Type	Constraint	Description	Values
genreID	INT	Primary Key	The unique identifier for the genre.	16
genreName	VARCHAR		The name of the genre.	Gospel

Customer & Employee Objects

❖ Personal Info

Field	Data Type	Constraint	Description	Values
personalInfoID	INT	Primary Key	The unique identifier for the personal information.	8
emailAddress	VARCHAR(120)	Unique Key	Email address	heatheresway@mac.com
password	VARCHAR(10)		Password	911ddc3b8f
firstName	VARCHAR(60)		The first name of the individual.	Heather
lastName	VARCHAR(60)		The last name of the individual.	Esway
dateOfBirth	DATE		The date of birth.	1998-07-25
gender	CHAR(1)		Describe whether the gender is male or female.	F

❖ Address

Field	Data Type	Constraint	Description	Values
addressID	INT	Primary Key	The unique identifier for the address	2
personalInfoID	INT	Foreign Key	The identifier for the personal information.	9
line1	VARCHAR(60)		The first line.	21 RosewooRd.
line2	VARCHAR(60)		The second line.	NULL
city	VARCHAR(40)		The name of the city.	Woodcliff Lake
state	VARCHAR(2)		The abbreviation of the state.	NJ
zipCode	VARCHAR(10)		Zip code.	07677
phone	VARCHAR(12)		Phone number.	201-653-4949
disabled	TINYINT(1)		Describe whether the address is disabled or not.	0

❖ Customer

Field	Data Type	Constraint	Description	Values
customerID	INT	Primary Key	The unique identifier for the customer.	6
personalInfoID	INT	Foreign Key	The identifier for the personal information.	11

❖ Employee

Field	Data Type	Constraint	Description	Values
employeeID	INT	Primary Key	The unique identifier for the employee.	4
personalIfoID	INT	Foreign Key	The identifier for the personal information.	4

Sales Objects

❖ Order

Field	Data Type	Constraint	Description	Values
orderID	INT	Primary Key	The unique identifier for the order.	3
customerID	INT	Foreign Key	The identifier for the customer.	1
orderDate	DATETIME		The date and time of the order.	2019-03-29 09:44:58
cardType	VARCHAR(50)		The type of the card: VISA, MASTER etc.	Visa
cardNumber	CHAR(16)		The number of the card.	4111111111111111
cardExpire	CHAR(7)		The expire date of the card.	04/2023

❖ Order Item

Field	Data Type	Constraint	Description	Values
itemID	INT	Primary Key	The unique identifier for the order item.	4
orderID	INT	Foreign Key	The identifier for the order.	4
trackID	INT	Foreign Key	The identifier for the track.	5
itemPrice	DECIMAL(10,2)		The price of the item.	1.50

Returns Objects

- Order Item

Field	Data Type	Constraint	Description	Values
returnID	INT	Primary Key	The unique identifier for the return.	2
itemID	INT	Foreign Key	The identifier for the order item.	3
dateOfReturn	DATE		Describe when the return was executed.	2019-04-05

Query Test and Verification

❑ Common Questions

1. How many customers does the store have?

2. The top sold products and least sold products over a week.

3. The average price of products in the same category (for example, rock, pop, country, hip-hop).

4. List customers by surname, date of birth and then sort them.

5. Clients that bought an average of four products in the last month.

6. List how many distinct albums each singer has.

7. List how many tracks are currently available for online streaming on each singer

❑ Questions Asked by the Client

1. Grasp the trend of music published in the past three years

2. What are the songs returned this month and when, which employees have taken the tasks?

3. Get customers' overall status of order with their background information

- ❖ Sample music data (numbers): 10 albums, 21 tracks, 20 artists, 20 genres

Query Statements for Verification 1

```
-- The Common Questions:
-- 1. How many customers does the store have?
SELECT COUNT(customer_id) AS number_of_cusomters
FROM customers;

-- the list of customers
SELECT customer_id, concat_ws(' ', first_name, last_name) AS customer_name
FROM customers
    JOIN personal_info ON customers.personal_info_personal_info_id = personal_info_id;


-- 2. The top sold products and least sold products over a week.
SELECT tracks_track_id, track_name, COUNT(tracks_track_id) AS number_of_sold
FROM order_items
    JOIN orders ON orders_order_id = order_id
    JOIN tracks ON tracks_track_id = track_id
WHERE order_date BETWEEN '2019-03-28' AND DATE_ADD('2019-03-28', INTERVAL 7 DAY)
GROUP BY tracks_track_id
ORDER BY number_of_sold DESC;

-- The list of sold tracks
SELECT tracks_track_id, track_name
FROM order_items
    JOIN orders ON orders_order_id = order_id
    JOIN tracks ON tracks_track_id = track_id
WHERE order_date BETWEEN '2019-03-28' AND DATE_ADD('2019-03-28', INTERVAL 7 DAY);


-- 3. The average price of products in the same category (for example, rock, pop, country, hip-hop).
SELECT genre_id, genre_name, ROUND(AVG(item_price), 2) AS average_price
FROM tracks
    JOIN order_items ON tracks_track_id = track_id
    JOIN genres ON genres_genre_id = genre_id
GROUP BY genres_genre_id;
```

Query Statements for Verification 2

```
-- 4. List customers by surname, date of birth and then sort them.
SELECT customer_id, last_name, date_of_birth
FROM customers
    JOIN personal_info ON personal_info_personal_info_id = personal_info_id
ORDER BY last_name, date_of_birth;


-- 5. Clients that ordered great than or equal to four products in the last month.
SELECT concat_ws(' ', first_name, last_name) AS client_name, COUNT(item_id) AS total_amount
FROM order_items
    JOIN orders ON orders_order_id = order_id
    JOIN customers ON customers_customer_id = customer_id
    JOIN personal_info ON personal_info_personal_info_id = personal_info_id
WHERE MONTH(order_date) = MONTH(CURDATE()) - 1
GROUP BY customer_id
HAVING total_amount >= 4;

-- The individual order list with the customers
SELECT concat_ws(' ', first_name, last_name) AS client_name,
    track_name AS ordered_track,
    item_price,
    DATE(order_date) AS ordered_date
FROM order_items
    JOIN tracks ON tracks_track_id = track_id
    JOIN orders ON orders_order_id = order_id
    JOIN customers ON customers_customer_id = customer_id
    JOIN personal_info ON personal_info_personal_info_id = personal_info_id
WHERE MONTH(order_date) = MONTH(CURDATE()) - 1;
```

Query Statements for Verification 3

```
-- 6. List how many distinct albums each singer has.
SELECT artist_name, COUNT(DISTINCT album_id) AS number_of_album
FROM albums
    JOIN artists ON artists_artist_id = artist_id
GROUP BY artists_artist_id;

-- The list of artists with the album
SELECT artist_name, album_id, album_title
FROM albums
    JOIN artists ON artists_artist_id = artist_id
ORDER BY artist_name;


-- 7. List how many tracks are currently available for online streaming on each singer
SELECT artists_artist_id, artist_name, COUNT(DISTINCT track_id) AS number_of_track
FROM tracks
    JOIN artists ON artists_artist_id = artist_id
GROUP BY artists_artist_id;

-- The list of available tracks on each album
SELECT album_title, track_name AS available_tracks
FROM tracks
    JOIN albums ON albums_album_id = album_id
ORDER BY album_title;

-- The list of available tracks on each album
SELECT album_title, COUNT(track_name) AS number_of_available_track
FROM tracks
    JOIN albums ON albums_album_id = album_id
GROUP BY album_title;
```

Query Statements for Verification 4

```
-- The Project Specific Questions:
-- 1. Grasp the trend of music published in the past three years
SELECT genre_id, genre_name, COUNT(DISTINCT track_id) AS number_of_track
FROM tracks
    JOIN genres ON genres_genre_id = genre_id
    JOIN albums ON albums_album_id = album_id
WHERE release_year BETWEEN 2016 AND 2018
GROUP BY genre_id
ORDER BY number_of_track DESC;


-- 2. What are the songs returned this month and when, which employees have taken the tasks?
SELECT return_id, track_name, concat_ws(' ', first_name, last_name) AS employee_name, date_of_return
FROM returns
    JOIN order_items ON order_items_item_id = item_id
    JOIN tracks ON tracks_track_id = track_id
    JOIN employees_has_returns ON returns_return_id = return_id
    JOIN employees ON employees_employee_id = employee_id
    JOIN personal_info ON personal_info_personal_info_id = personal_info_id
WHERE date_of_return BETWEEN '2019-04-01' AND CURDATE();


-- 3. Get customers' overall status of order with their background information
SELECT concat_ws(' ', first_name, last_name) AS customer_name, (YEAR(CURDATE()) - YEAR(date_of_birth)) AS age, gender,
    city AS residence_area, COUNT(genre_name) AS number_of_order, genre_name, order_date
FROM customers
    JOIN personal_info ON customers.personal_info_personal_info_id = personal_info_id
    JOIN addresses ON addresses.personal_info_personal_info_id = personal_info_id
    JOIN orders ON customers_customer_id = customer_id
    JOIN order_items ON orders_order_id = order_id
    JOIN tracks ON tracks_track_id = track_id
    JOIN genres ON genres_genre_id = genre_id
GROUP BY customer_name, genre_name WITH ROLLUP;
```