

CS 340 - INTRODUCTION TO DATABASES

Album Ranking and Sales Database

FINAL SUBMISSION

Group 60

Caleb Richter & Andrew Walsh

Project URL: <http://classwork.engr.oregonstate.edu:6896/>

December 2025

TABLE OF CONTENTS

1. Executive Summary of Feedback and Changes
2. Project Overview
3. Database Outline
4. Entity-Relationship Diagram (ERD)
5. Schema Diagram
6. Sample Data
7. UI Screen Captures with CRUD Operations
8. Citations

1. EXECUTIVE SUMMARY: FEEDBACK AND CHANGES

Group 60's database management system for McFunky Rhythm's Records evolved through five iterations from October through December 2025, growing into a full-stack application with seven entities, two M:N relationships with complete CRUD, and eighteen stored procedures. TA and peer feedback corrected critical design flaws including inconsistent naming conventions, inappropriate data types, and missing CASCADE declarations. Feedback also drove impactful changes such as replacing text inputs with dropdown menus for Artist/Genre selection, eliminating foreign key constraint errors and improving data integrity at the UI level. Additional refinements included customer name parsing, NULL customer support in Sales transactions, and associated line item management within expandable sales records.

Full disclosure here – we used AI for a lot of this project since we were allowed to under the assumption that we cited all usage, did not mindlessly use whatever AI generated, understood what it was doing, and generally was responsible with using it as a tool and not a replacement of “knowledge”. We initially developed our SQL DDL and DML files and Albums frontend/backend functionality independently using knowledge gained from CS340 modules, readings, walkthroughs, and/or templates before engaging AI for remaining components, iteratively improving upon them until completed to our satisfaction. While the project requirements were achievable using course materials alone—we viewed AI usage as a valuable learning opportunity to study and understand complex SQL, React, and JavaScript code patterns above and beyond the basic requirements.

AI tools (Claude 3.5 Sonnet and ChatGPT) accelerated development through CRUD component generation, stored procedure templates, and debugging assistance. Without extensive React experience, we were able to freely explore complex frontend concepts such as component composition, routing architecture, and state management through iterative dialogue with Claude. Similarly, Claude generated sophisticated SQL stored procedures, such as the Sales creation procedure with complex transaction handling, which would have required significant research and trial-and-error to develop independently. However, limitations did emerge. AI failed at times to catch validation bugs, maintain cross-file consistency, and required extensive manual refinement for transaction management and business logic, just to name a few. AI excelled at isolated, well-specified tasks but required human oversight for system-wide coherence and validation.

The final implementation exceeds requirements with seven entities (four required), calculated fields, database reset functionality, and natural key lookups. Foreign key dropdowns prevent constraint violations. LEFT JOIN queries handle NULL customers gracefully. These features demonstrate mature database design solving real-world business problems. The iterative approach—where TA feedback ensured technical correctness, peer feedback improved usability, and AI tools accelerated implementation—produced a robust solution validating AI as a productivity multiplier requiring critical human oversight in database engineering.

2. PROJECT OVERVIEW

PROBLEM STATEMENT:

Over the last five years, McFunky Rhythm's Records, a local record store, has averaged only 50 sales per month and typically only manages to just break even. They would like to increase this number to at least 150 sales a month to justify keeping their doors open.

As the store originally opened in the 1990s, the owners typically only have records from that generation of artists and earlier. Feedback from their customers indicate that the owners are "out-of-touch" and the albums they have had on display for months are not selling because "not everyone likes dad rock" and "there's no music from TikTok here". Additionally, the owners have developed arthritis from maintaining handwritten records and are seeking a digital solution.

PROPOSED SOLUTION:

A centralized, transactional database capturing Artists, Albums, Album Ratings, Customers, Sales, LineItems, and Genres that is capable of storing:

- 20,000 albums
- 5,000 artists
- 20,000 customers
- 50,000 ratings
- 10,000 sales
- 50,000 line items

By systematically recording sales transactions, customer information, album and artist information, and album ratings, the database will:

- Manage day-to-day operations (inventory management and sales)
- Provide a data-driven foundation for decision-making
- Eliminate manual record-keeping and reduce arthritis flare-ups
- Enable analysis of customer preferences and trending albums
- Support informed inventory purchasing decisions

The database has sufficient capacity to accommodate at least one year of transactional records before archival. Additional space can be allocated as business grows.

3. DATABASE OUTLINE

ENTITY: Artists

Purpose: Records the details of artists/bands

Attributes:

- artistID: VARCHAR(50), Primary Key, Not-Null, Unique
- description: VARCHAR(255), Not-Null

Relationships:

- 1:M with Albums (one artist can have multiple albums)

Notes:

- artistID is editable as it serves as a natural key (band/artist name)
- Collaboration albums treated as separate artist entries

ENTITY: Genres

Purpose: Records music genres (reference data)

Attributes:

- genreID: VARCHAR(50), Primary Key, Not-Null, Unique
- description: VARCHAR(255), Not-Null

Relationships:

- 1:M with Albums (one genre can have multiple albums)

Notes:

- Read-only entity (no CUD operations)
- Each album classified under single genre to avoid overcomplication

ENTITY: Albums

Purpose: Records the details of albums in inventory

Attributes:

- albumID: INT, Primary Key, Not-Null, Auto-Incremented, Unique
- albumName: VARCHAR(100), Not-Null
- albumPrice: DECIMAL(4,2), Not-Null
- amountInStock: INT, Not-Null, Default(0)
- artistID: VARCHAR(50), Foreign Key, ON DELETE NO ACTION
- genreID: VARCHAR(50), Foreign Key, ON DELETE NO ACTION

Calculated Fields (not stored):

- avgRating: DECIMAL(2,1) - Average rating from AlbumRatings table

Calculated as: ROUND(AVG(AlbumRatings.albumRating), 2)

NULL if no ratings exist

Relationships:

- M:1 with Artists (many albums belong to one artist)
- M:1 with Genres (many albums belong to one genre)
- M:N with Sales via LineItems intersection table
- M:N with Customers via AlbumRatings intersection table

Notes:

- albumPrice is current price (may differ from historical sale prices)
- avgRating calculated dynamically to prevent data anomalies
- ON DELETE NO ACTION prevents deletion if dependencies exist

ENTITY: Customers

Purpose: Records the details of customers who make purchases

Attributes:

- customerID: INT, Primary Key, Not-Null, Auto-Incremented, Unique
- firstName: VARCHAR(45), Not-Null
- lastName: VARCHAR(45), Not-Null
- phoneNumber: VARCHAR(15), Default NULL
- email: VARCHAR(50), Default NULL

Display Fields (calculated):

- customer: CONCAT(firstName, ' ', lastName) - Full name for display

Relationships:

- 1:M with Sales (one customer can make multiple sales)
- 1:M with AlbumRatings (one customer can rate multiple albums)
- M:N with Albums via AlbumRatings intersection table

Notes:

- firstName/lastName stored separately in database
- Full name displayed as concatenated field in UI
- Name parsing implemented in UI for single-input updates
- phoneNumber and email optional (can be NULL)

ENTITY: Sales

Purpose: Records the details of sales transactions

Attributes:

- salesID: INT, Primary Key, Not-Null, Auto-Incremented, Unique
- customerID: INT, Foreign Key, ON DELETE SET NULL
- totalCost: DECIMAL(10,2), Not-Null

- purchaseDate: DATE, Not-Null, Default CURRENT_DATE

Relationships:

- M:1 with Customers (many sales belong to one customer)
- M:N with Albums via LineItems intersection table

Notes:

- customerID can be NULL (allows customer deletion without orphaning sales)
- totalCost stored explicitly as critical business metric
- LEFT JOIN used to display '<customer deleted>' for NULL customers
- No DELETE operation (sales records preserved for accounting)

ENTITY: AlbumRatings (Intersection Table)

Purpose: Records customer ratings of albums (M:N between Albums and Customers)

Attributes:

- albumRatingID: INT, Primary Key, Not-Null, Auto-Incremented, Unique
- albumRating: DECIMAL(2,1), Not-Null (0.0 to 5.0 scale)
- albumID: INT, Foreign Key, ON DELETE CASCADE
- customerID: INT, Foreign Key, ON DELETE CASCADE

Relationships:

- M:1 with Albums (many ratings for one album)
- M:1 with Customers (many ratings by one customer)

Notes:

- Facilitates M:N relationship between Albums and Customers
- Multiple ratings allowed per album/customer combination
- ON DELETE CASCADE removes ratings if album or customer deleted
- Used to calculate Albums.avgRating dynamically

ENTITY: LineItems (Intersection Table)

Purpose: Records line items in sales transactions (M:N between Sales and Albums)

Attributes:

- lineItemID: INT, Primary Key, Not-Null, Auto-Incremented, Unique
- quantity: INT, Not-Null
- albumPrice: DECIMAL(4,2), Not-Null
- salesID: INT, Foreign Key, ON DELETE CASCADE
- albumID: INT, Foreign Key, ON DELETE NO ACTION

Calculated Fields (not stored):

- lineItemTotal: quantity * albumPrice

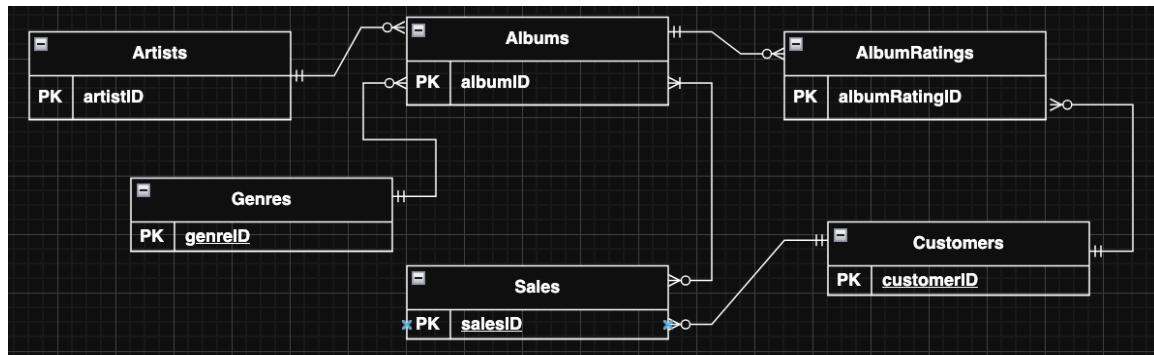
Relationships:

- M:1 with Sales (many line items in one sale)
- M:1 with Albums (many line items reference one album)

Notes:

- Facilitates M:N relationship between Sales and Albums
- albumPrice captured at time of sale (may differ from current Albums.albumPrice)
- Critical for historical accuracy as prices fluctuate
- ON DELETE CASCADE removes line items if sale deleted
- ON DELETE NO ACTION prevents album deletion if in line items
- lineItemTotal calculated dynamically (quantity * albumPrice)

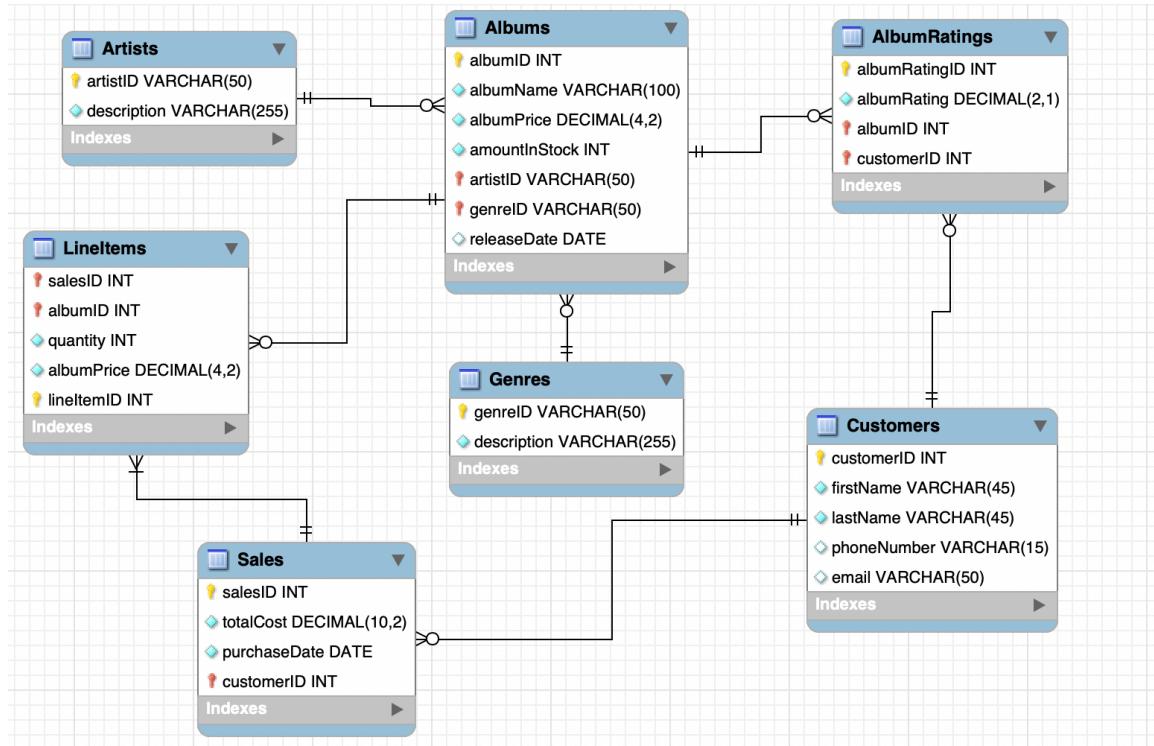
4. ENTITY-RELATIONSHIP DIAGRAM (ERD)



Key Relationships:

- 1:M - Artists to Albums
- 1:M - Genres to Albums
- M:N - Albums to Sales (via LineItems)
- M:N - Albums to Customers (via AlbumRatings)
- 1:M - Customers to Sales
- 1:M - Customers to AlbumRatings

5. SCHEMA DIAGRAM



Schema should show physical database structure with:

- All 7 tables (Artists, Genres, Albums, Customers, Sales, AlbumRatings, LineItems)
- Primary keys (PK) clearly marked
- Foreign keys (FK) clearly marked
- All attributes with data types
- Relationship lines showing 1:M and M:N connections
- ON DELETE actions labeled (CASCADE, SET NULL, NO ACTION)

6. SAMPLE DATA

Artists Sample Data

| artistID | description |
|------------|-----------------------------------|
| Sublime | Classic Reggae / Ska-punk band |
| MotleyCrue | Classic 80s glam rock band |
| Metallica | Classic Metal / Thrash metal band |
| Beatles | 60s Rock n Roll |

Genres Sample Data

| genreID | description |
|-----------|--|
| Rock | Classic & Hard Rock |
| Metal | Heavy Metal / Thrash Metal / Classic Metal |
| Reggae | Reggae / Ska |
| RockNRoll | 50s - 70s Rock and Roll |

Albums Sample Data

| albumID | albumName | albumPrice | amountInStock | artistID | genreID |
|---------|-------------------|------------|---------------|------------|-----------|
| 1 | Sublime | 19.99 | 6 | Sublime | Reggae |
| 2 | Dr. Feelgood | 21.99 | 4 | MotleyCrue | Rock |
| 3 | Master Of Puppets | 24.99 | 5 | Metallica | Metal |
| 4 | Abbey Road | 22.50 | 7 | Beatles | RockNRoll |

Customers Sample Data

| customerID | firstName | lastName | phoneNumber | email |
|------------|-----------|----------|--------------|--------------------|
| 1 | Jimmy | Buffet | 123-321-1234 | j.buffet@gmail.com |
| 2 | Steve | Vaughn | 901-801-7012 | s.vaughn@aol.com |
| 3 | Alice | Johnson | NULL | NULL |
| 4 | Janice | Jackson | 555-666-7777 | jj@hotmail.com |

Sales Sample Data

| salesID | customerID | totalCost | purchaseDate |
|---------|------------|-----------|--------------|
| 1 | 1 | 42.49 | 2025-10-30 |
| 2 | 2 | 19.99 | 2025-10-16 |
| 3 | 3 | 24.99 | 2025-10-11 |

AlbumRatings Sample Data

| albumRatingID | albumRating | albumID | customerID |
|---------------|-------------|---------|------------|
| 1 | 4.5 | 1 | 1 |
| 2 | 2.5 | 2 | 2 |
| 3 | 1.0 | 3 | 1 |
| 4 | 3.7 | 4 | 3 |

LineItems Sample Data

| lineItemID | quantity | albumPrice | salesID | albumID |
|------------|----------|------------|---------|---------|
| 1 | 1 | 19.99 | 1 | 1 |
| 2 | 1 | 22.50 | 1 | 4 |
| 3 | 1 | 19.99 | 2 | 1 |
| 4 | 1 | 24.99 | 3 | 3 |

7. UI SCREEN CAPTURES WITH CRUD OPERATIONS

HOME PAGE - RESET

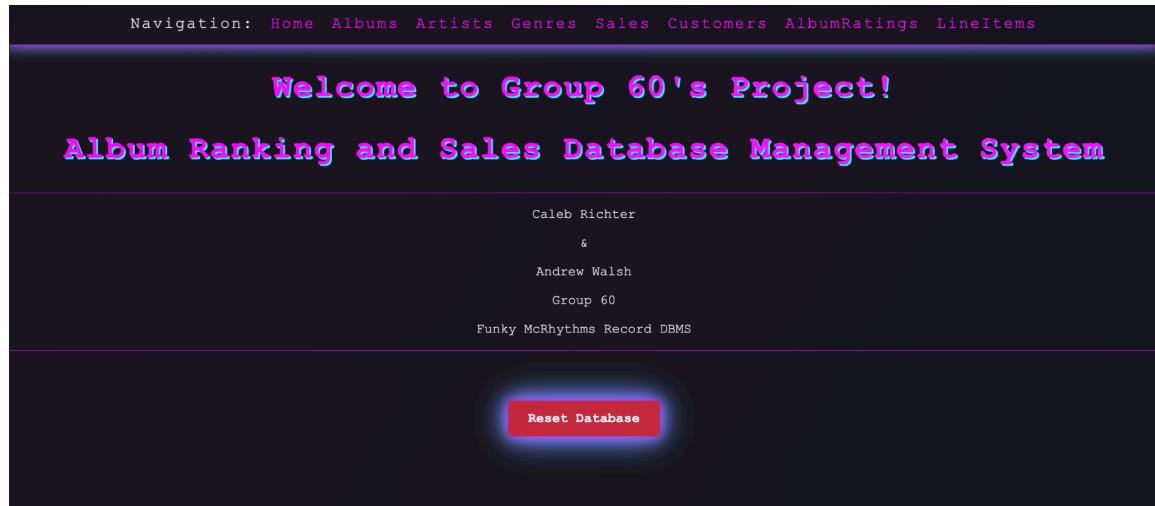
Navigation: Home Albums Artists Genres Sales Customers AlbumRatings LineItems

Welcome to Group 60's Project!

Album Ranking and Sales Database Management System

Caleb Richter
&
Andrew Walsh
Group 60
Funky McRhythms Record DBMS

[Reset Database](#)



ALBUMS - CREATE/READ/UPDATE/DELETE

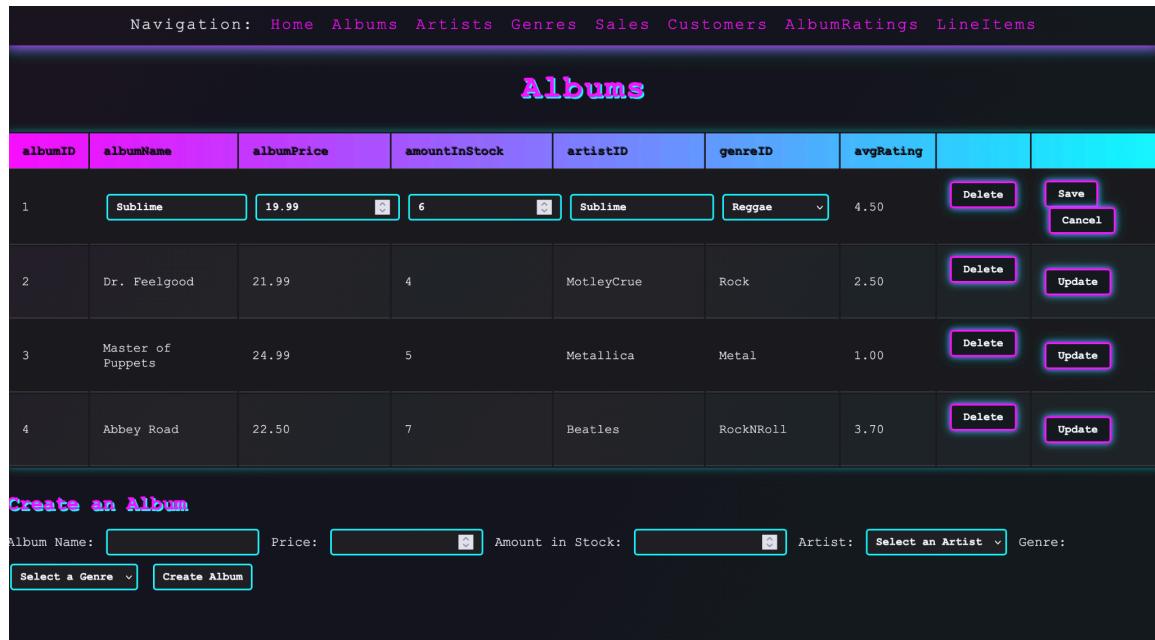
Navigation: Home Albums Artists Genres Sales Customers AlbumRatings LineItems

Albums

| albumID | albumName | albumPrice | amountInStock | artistID | genreID | avgRating | | | |
|---------|-------------------|------------|---------------|------------|-----------|-----------|------------------------|------------------------|--|
| 1 | Sublime | 19.99 | 3 | 6 | Sublime | Reggae | 4.50 | Delete | Save Cancel |
| 2 | Dr. Feelgood | 21.99 | 4 | MotleyCrue | Rock | 2.50 | Delete | Update | |
| 3 | Master of Puppets | 24.99 | 5 | Metallica | Metal | 1.00 | Delete | Update | |
| 4 | Abbey Road | 22.50 | 7 | Beatles | RockNRoll | 3.70 | Delete | Update | |

Create an Album

Album Name: Price: Amount in Stock: Artist: Genre:



ARTISTS - CREATE/READ/UPDATE/DELETE

Navigation: Home Albums Artists Genres Sales Customers AlbumRatings LineItems

Artists

| artistID | description | | |
|------------|-----------------------------------|--------|-------------|
| Beatles | 60s Rock n Roll | Delete | Save Cancel |
| Metallica | Classic Metal / Thrash metal band | Delete | Update |
| MotleyCrue | Classic 80s glam rock band | Delete | Update |
| Sublime | Classic Reggae / Ska-punk band | Delete | Update |

Create an Artist

Artist: Description:

GENRES - READ

Navigation: Home Albums Artists Genres Sales Customers AlbumRatings LineItems

Genres

| genreID | description |
|-----------|--|
| Metal | Heavy Metal / Thrash Metal / Classic Metal |
| Reggae | Reggae / Ska |
| Rock | Classic & Hard Rock |
| RockNRoll | 50 - 70s Rock and Roll |

SALES - CREATE/READ/UPDATE/DELETE

Navigation: Home Albums Artists Genres Sales Customers AlbumRatings LineItems

Sales

| salesID | customer | totalCost | purchaseDate | | |
|---------|---------------|---|--|--|---|
| 1 | Jimmy Buffet | 42.49 <input type="button" value="Edit"/> | 10 / 30 / 2025 <input type="button" value="Edit"/> | <input type="button" value="Show Line Items"/> | <input type="button" value="Save"/> <input type="button" value="Cancel"/> |
| 2 | Steve Vaughn | \$19.99 | 2025-10-16 | <input type="button" value="Show Line Items"/> | <input type="button" value="Update"/> |
| 3 | Alice Johnson | \$24.99 | 2025-10-11 | <input type="button" value="Show Line Items"/> | <input type="button" value="Update"/> |

Create a Sale

Customer First Name: Customer Last Name: Sale Date:

Line Items

Album: Price: Quantity:

SALES - CREATE/READ/UPDATE/DELETE ASSOCIATED LINE ITEMS

| Sales | | | | | | |
|---|---------------|---|---------------------------------------|---------------------------------|------------------------|---|
| salesID | customer | totalCost | purchaseDate | | | |
| 1 | Jimmy Buffet | \$42.49 | 2025-10-30 | Show Line Items | | Update |
| lineItemID | albumName | albumPrice | quantity | lineItemTotal | | |
| 1 | Sublime | 19.99 <input type="button" value="Edit"/> | 1 <input type="button" value="Edit"/> | \$19.99 | Delete | Save Cancel |
| 2 | Abbey Road | \$22.50 | 1 | \$22.50 | Delete | Update |
| Add Item: <input type="button" value="Select Album"/> <input type="text" value="Price"/> <input type="text" value="Quantity"/> <input type="button" value="Add"/> | | | | | | |
| 2 | Steve Vaughn | \$19.99 | 2025-10-16 | Show Line Items | | Update |
| 3 | Alice Johnson | \$24.99 | 2025-10-11 | Show Line Items | | Update |

CUSTOMERS - CREATE/READ/UPDATE/DELETE

Navigation: Home Albums Artists Genres Sales Customers AlbumRatings LineItems

Customers

| customerID | customer | phoneNumber | email | | |
|------------|----------------|--------------|--------------------|------------------------|---|
| 1 | Jimmy Buffet | 123-321-1234 | j.buffet@gmail.com | Delete | Save Cancel |
| 2 | Steve Vaughn | 901-801-7012 | s.vaughn@aol.com | Delete | Update |
| 3 | Alice Johnson | | | Delete | Update |
| 4 | Janice Jackson | 555-666-7777 | jj@hotmail.com | Delete | Update |

Create a Customer

First Name: Last Name: Phone Number: Optional Email: Optional [Create Customer](#)

ALBUM RATINGS - CREATE/READ/UPDATE/DELETE

Navigation: Home Albums Artists Genres Sales Customers AlbumRatings LineItems

Album Ratings

| albumRatingID | albumRating | albumName | customer | | |
|---------------|-------------|-------------------|---------------|------------------------|---|
| 1 | 4.5 | Sublime | Jimmy Buffet | Delete | Save Cancel |
| 2 | 2.5 | Dr. Feelgood | Steve Vaughn | Delete | Update |
| 3 | 1.0 | Master of Puppets | Jimmy Buffet | Delete | Update |
| 4 | 3.7 | Abbey Road | Alice Johnson | Delete | Update |

Create an Album Rating

Album Name: Customer Name: Rating: Select a Rating [Create Rating](#)

LINE ITEMS - READ (full CRUD on Sales page)

| Navigation: Home Albums Artists Genres Sales Customers AlbumRatings LineItems | | | | | |
|---|---------|-------------------|------------|----------|---------------|
| All Line Items | | | | | |
| lineItemID | salesID | albumName | albumPrice | quantity | lineItemTotal |
| 1 | 1 | Sublime | 19.99 | 1 | 19.99 |
| 3 | 2 | Sublime | 19.99 | 1 | 19.99 |
| 4 | 3 | Master of Puppets | 24.99 | 1 | 24.99 |
| 2 | 1 | Abbey Road | 22.50 | 1 | 22.50 |

8. CITATIONS

AI TOOL CITATIONS

Citation 1

AI Model: Claude AI

Date: 11/13/2025

Purpose: Updated TableRow.jsx and GenericUpdateButton.jsx with in-place editing functionality

Prompts:

- Prompt 1: Implement an update button to make row values editable in place
- Prompt 2: Making first attribute/column non-editable

Source URL: <https://claude.ai>

Citation 2

AI Tools: GitHub Copilot / ChatGPT

Date: 11/06/2025

Purpose: Debugging deprecated VITE version errors by analyzing console log errors. AI provided insight into pathing bug caused by old VITE version on engineering server, though solution required manual modification.

Citation 3

AI Model: ChatGPT

Date: 11/13/2025

Purpose: Generated 80s-themed template for index.css stylesheet, which was then modified

Prompt: "Please make a globally style sheet for REACT index.css sheet that is 80s themed."

Source URL: <https://chatgpt.com>

Citation 4

AI Model: Claude 3.5 Sonnet

Date: 11/20/2024

Purpose: Created delete functionality connecting React frontend to Node.js backend. Generated AlbumDeleteButton.jsx component, AlbumTableRow.jsx component, /Albums/delete route in server.js, and sp_DeleteAlbum stored procedure. Fixed routing issues causing NULL values.

Source URL: <https://claude.ai/>

Citation 5

AI Model: Claude 3.5 Sonnet

Date: 11/20/2024

Purpose: Created database reset functionality. Generated ResetButton.jsx component with fetch request to backend /reset-database route, confirmation dialog, error handling, and callback for post-reset actions.

Source URL: <https://claude.ai/>

Citation 6

AI Model: Claude 3.5 Sonnet

Date: 11/20/2024

Purpose: Integrated ResetButton component into Home page. Added import statement, backend URL configuration, callback handler, and positioned reset button at bottom of homepage.

Source URL: <https://claude.ai/>

Citation 7

AI Model: Claude 3.5 Sonnet

Date: 12/04/2025

Purpose: Comprehensive implementation of remaining CRUD operations including Sales UPDATE, Albums avgRating calculation, Customer name parsing in locations, AlbumRatingTableRow.jsx, CreateAlbumRatingsForm.jsx, CustomerTableRow.jsx and CustomerUpdateButton.jsx updates, SalesUpdateButton.jsx, button styling consistency, SQL documentation, and DML query updates. Source URL: <https://claude.ai/>

Citation 8

AI Model: Claude 3.5 Sonnet

Date: 12/07/2025

Purpose: Updated CreateAlbumForm.jsx to use dropdown menus for Artist and Genre selection to prevent foreign key constraint errors. Updated group60_DML.sql documentation.

Files Modified: Frontend/src/Components/AlbumComponents/CreateAlbumForm.jsx, Backend/database/group60_DML.sql

Source URL: <https://claude.ai/>

Citation 9

AI Model: Claude 3.5 Sonnet (Claude Sonnet 4)

Date: 12/08/2025

Purpose: Created final submission documentation package including comprehensive Word document (Group60_Final_Submission.docx) with cover page, table of contents, executive summary, complete database outline for all entities, ERD and Schema sections with placeholders, formatted sample data tables, screenshot placeholders, and complete citations section. Updated README.md with all consolidated citations. Validated DML-to-stored procedures alignment across all entities. Fixed customer name validation bugs in CustomerUpdateButton.jsx and AlbumRatingTableRow.jsx requiring both first and last name. Identified and resolved database reset error caused by customerId NOT NULL/ON DELETE SET NULL contradiction. Verified DDL-to-outline consistency and confirmed sample data accuracy.

Created comprehensive submission guide with troubleshooting, screenshot best practices, and final checklist. Condensed executive summary to one-page format meeting submission requirements. Developed proper executive summary meeting assignment requirements: narrative format rather than changelog structure, comprehensive AI strengths/weaknesses analysis, reflection on how feedback influenced design decisions, and manager-level perspective. Condensed executive summary through multiple iterations to fit one-page requirement (~280 words final).

Files Created/Modified:

- Group60_Final_Submission.docx (complete submission document)
- README.md (updated with citations)
- CustomerUpdateButton.jsx (name validation fix)
- AlbumRatingTableRow.jsx (parameter passing fix)

Source URL: <https://claude.ai/>

NON-AI CITATIONS

Citation 10

Source: CS340 Modules/Explorations

Date: November 2025

Purpose: Utilized REACT template code and video walkthroughs for several of the generic components/pages, CRUD operations, database connection, and project structure.

Source URL: <https://canvas.oregonstate.edu/courses/2017561/pages>

END OF FINAL SUBMISSION DOCUMENT

Submitted by:

Caleb Richter & Andrew Walsh

Group 60

CS 340 - INTRODUCTION TO DATABASES

Oregon State University

December 2025