Project Name: Bookstore Management System

Subtitle: Oracle SQL Database Design and Implementation

Professor: Abe F. Kazemzadeh
Designed by: Syed Hassan Ahmed
Class: SEIS 630-01 DBMS & Design,Fall2024
Date: 12/08/2024

Agenda

S. No.	Topic Name	Slides	Slide Number
1	**Reason for Choosing the Topic**	1	1
2	**Application of Class Learning**	3	2-4
3	**Challenges Encountered**	1	5
4	**Lessons Learned**	2	6-7
5	**Future Work**	1	8

Reason for Choosing the Topic

- **Why Bookstore Management System?**
- A large percentage of people in the USA incorporate reading into their daily routines, leading to the need for a streamlined solution for bookstore management.

- **Personal Motivation**:
- Curious about improving bookstore operations, from inventory management to customer order tracking.

- **Project Goal**:
- To develop an efficient database solution for managing books, authors, customers, and orders with scalability.

Application of Class Learning – Database Design

- **Class Concept Applied**:

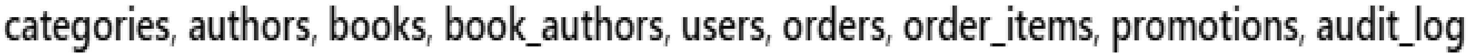
Designed the **Bookstore Database** using Oracle SQL, incorporating concepts like **ER modeling**, **foreign keys**, **triggers**, and **constraints**.

- **ER Diagram**:

Used **Oracle SQL Developer** for visualizing the database schema. Defined relationships between tables such as **1:M** (books and categories), **M:N** (books and authors) and **more**..

- **Entity Relationships**:

Created tables like **Books**, **Categories**, **Orders**, **Users**, **Promotions** and **more**, implementing real-world business logic in the structure.



categories, authors, books, book_authors, users, orders, order_items, promotions, audit_log

Application of Class Learning – SQL Queries

- **Joins and Data Retrieval**:

Used **SQL JOINs** to retrieve complex data from multiple tables. Example: retrieving books along with their authors and categories.

- **Triggers and Automation**: (Show examples)

Implemented **triggers** to automatically log changes (INSERT, UPDATE, DELETE) in the **audit_log** table for tracking book (books table) data modifications.

- **Constraints for Data Integrity**:

Applied **CHECK constraints** for validating inputs, such as ensuring non-negative stock quantities and valid discount percentages.

Application of Class Learning – Data Manipulation

- **Inserting Data**:

Populated tables with realistic data using **Mockaroo**, simulating books, authors, users, promotions and so on.

- **Data Modifications**:

Implemented **INSERT**, **UPDATE**, and **DELETE** queries to manipulate data, following the design of relationships between entities (e.g., books, orders).

- **Sample Query Execution**: (Show examples)

Examples of queries executed:

1. Top-Selling Books by Quantity
2. Most Frequent Customers
3. Revenue Generated by Category
4. Books with Active Promotions
5. Best-Selling Books with Revenue and Author Info

Challenges Encountered – Trigger and Data Integrity

- **Triggers**:

Learning to write and debug triggers to maintain audit logs after data changes (insert, update, delete) in the **Books** table was complex.

- **Data Integrity**:

Ensuring proper **foreign key** constraints and **check constraints** to maintain referential integrity across multiple tables was an important challenge.

- **Database Optimization:**

Ensuring performance optimization with large datasets and efficient querying was a challenge.

- **Integrating SQL with Django**:

Initially planned to use **Django** for front-end development, but encountered difficulties in integrating the database and maintaining data integrity.

- **Complex Relationships**:

Handling **many-to-many** relationships between books and authors proved to be more complex than initially anticipated.

Lessons Learned – Database Design and Structure

- **Designing Relational Databases**:
- Learned how to build a fully functional relational database with normalized tables and relationships (1:1, 1:N, M:N).

- **Database Efficiency**:
- Optimized data retrieval with **joins**, **indexes**, and **proper query planning** to ensure performance under increasing data loads.

- **Audit Logging and Triggers**:
- Gained deep insight into using **triggers** to automate actions, such as logging database changes and maintaining historical data.

Lessons Learned – SQL Programming

- **Improved SQL Querying**:
- Gained expertise in writing complex queries involving **aggregates**, **joins**, and **nested subqueries** to retrieve meaningful data from the database.

- **Data Integrity Practices**:
- Applied concepts like **CHECK constraints**, **foreign keys**, and **triggers** to maintain database integrity and avoid errors.

- **Handling Large Datasets**:
- Learned techniques to handle large datasets effectively, including using batch operations and optimizing query performance.

Future Work – Enhancements and Features

- **Front-end Integration**:

Future plans include integrating a **front-end interface** using **Django** or other web frameworks, making the system user-friendly for administrators and customers.

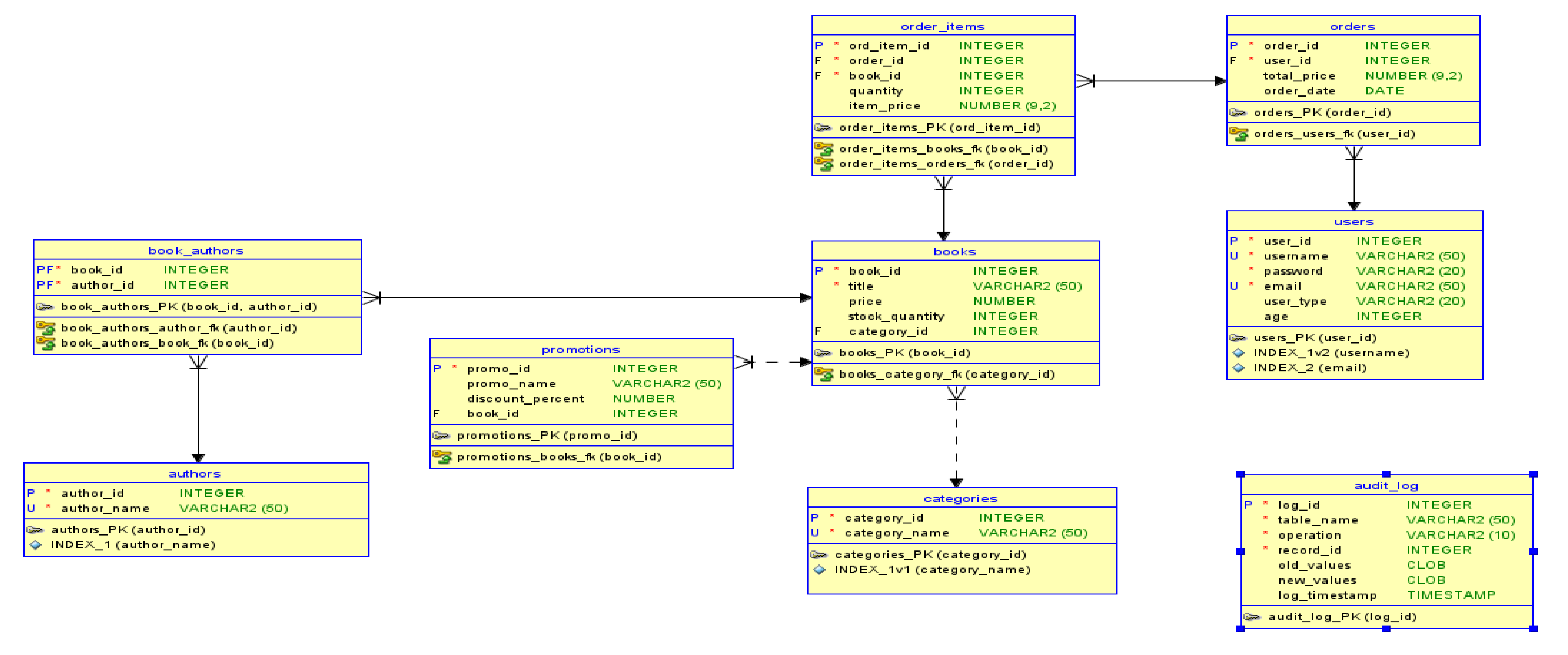
- **Advanced Reporting**:

Implement advanced reporting features to analyze book sales, user behavior, and promotions, generating insights for decision-making.

- **Recommendation System**:

Develop a **book recommendation system** based on user purchase history and ratings.

ER Diagram of Bookstore Database



order_items

P	*	ord_item_id	INTEGER
F	*	order_id	INTEGER
F	*	book_id	INTEGER
		quantity	INTEGER
		item_price	NUMBER (9,2)

order_items_PK (ord_item_id)

order_items_books_fk (book_id)
order_items_orders_fk (order_id)

orders

P	*	order_id	INTEGER
F	*	user_id	INTEGER
		total_price	NUMBER (9,2)
		order_date	DATE

orders_PK (order_id)

orders_users_fk (user_id)

users

P	*	user_id	INTEGER
U	*	username	VARCHAR2 (50)
	*	password	VARCHAR2 (20)
U	*	email	VARCHAR2 (50)
		user_type	VARCHAR2 (20)
		age	INTEGER

users_PK (user_id)
INDEX_1v2 (username)
INDEX_2 (email)

book_authors

PF	*	book_id	INTEGER
PF	*	author_id	INTEGER

book_authors_PK (book_id, author_id)

book_authors_author_fk (author_id)
book_authors_book_fk (book_id)

books

P	*	book_id	INTEGER
	*	title	VARCHAR2 (50)
		price	NUMBER
		stock_quantity	INTEGER
F		category_id	INTEGER

books_PK (book_id)

books_category_fk (category_id)

promotions

P	*	promo_id	INTEGER
		promo_name	VARCHAR2 (50)
		discount_percent	NUMBER
F		book_id	INTEGER

promotions_PK (promo_id)

promotions_books_fk (book_id)

authors

P	*	author_id	INTEGER
U	*	author_name	VARCHAR2 (50)

authors_PK (author_id)
INDEX_1 (author_name)

categories

P	*	category_id	INTEGER
U	*	category_name	VARCHAR2 (50)

categories_PK (category_id)
INDEX_1v1 (category_name)

audit_log

P	*	log_id	INTEGER
	*	table_name	VARCHAR2 (50)
	*	operation	VARCHAR2 (10)
	*	record_id	INTEGER
		old_values	CLOB
		new_values	CLOB
		log_timestamp	TIMESTAMP

audit_log_PK (log_id)

Thank you

Syed Hassan Ahmed

Stay connected with me on

https://www.**linkedin**.com/in/syed4

Gmail: hassan.tonk@gmail.com