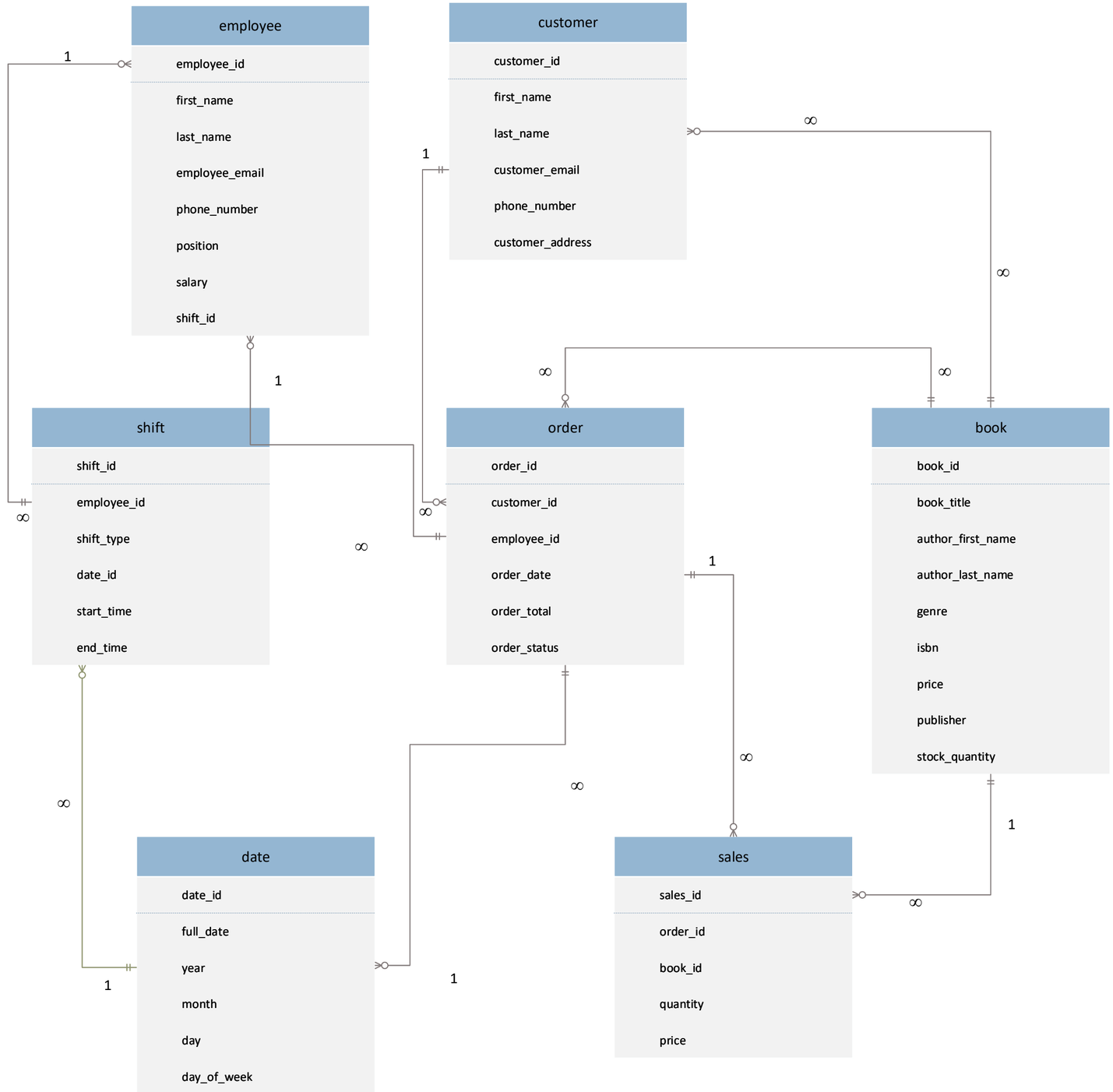
CALEN LAU
SQL ASSIGNMENT – SMALL BOOKSTORE
QUESTION 1 AND 2

Question 1 - Create a logical model for a small bookstore.
At the minimum it should have employee, order, sales, customer, and book entities (tables). Determine sensible column and table design based on what you know about these concepts.
Keep it simple, but work out sensible relationships to keep tables reasonably sized.

Question 2 - We want to create employee shifts, splitting up the day into morning and evening. Add this to the ERD.



SQL ASSIGNMENT
CALEN LAU
DUE SEPT 28, 2024

QUESTION 3

The store wants to keep customer addresses. Propose two architectures for the CUSTOMER_ADDRESS table, one that will retain changes, and another that will overwrite. Which is type 1, which is type 2? Hint, search type 1 vs type 2 slowly changing dimensions.
Bonus: Are there privacy implications to this, why or why not?

Of the two architectures for the CUSTOMER_ADDRESS table, the one that will retain changes is Type 1 Slowly Changing Dimensions (SCD), while Type 2 SCD is the architecture that will overwrite previous data.

Type 1 vs. Type 2 Slowly Changing Dimensions

Type 1 SCD – Retains Changes

This db architecture design involves creating a new record in the CUSTOMER_ADDRESS table whenever an address change occurs. This allows for the retention of historical data, which supports the ability to track changes that are made in the data.

Type 2 SCD – Overwrites

This architecture type overwrites the old address with the new address when a change is made. Therefore, only the current information is kept, and the history of previous addresses are not retained.

Proposed Architectures

Type 1 Architecture – Retains Changes

CUSTOMER_ADDRESS
customer_id
address
city
province
postalcode
start_date
end_date
is_current

Example of the display information for a customer's address information with Type 1 SCD:

customer_id	customer_address	city	province	postalcode	start_date	end_date	is_current
1	10 Lakeshore Rd	Oshawa	ON	L3Y3S7	2024-02-01	NULL	1
1	2 Parklawn Ave	Toronto	ON	M2R4G7	2016-05-30	2024-01-31	0

- A new record is created each time a customer's address changes.
- The start_date and end_date columns are used to determine the time range during which an address was active.
- The is_current flag is used for quick lookup of the current address.
- This approach maintains a full history of address changes for each customer.

Type 2 Architecture – Overwrites

CUSTOMER_ADDRESS
customer_id
address
city
province
postalcode
last_updated

Example of the display information for a customer's current address information with Type 2 SCD:

customer_id	address	city	province	postalcode	last_uodated
1	10 Lakeshore Rd	Oshawa	ON	L3Y3S7	2024-02-01

- This table stores only the most recent address for each customer.
- Any updates to a customer's address will overwrite the existing record.
- There is no historical tracking of previous addresses.

Privacy Implications

There are privacy implications to consider for both approaches.

For Type 1 architecture in which changes are retained, a higher risk for privacy breaches exist primarily because historical address data is stored; therefore, the more sensitive and personal information is retained, the greater the risk profile. As such, greater adherence to Canadian privacy regulations in the handling of personal data is recommended, which includes corporate policies to determine hierarchies of access controls by company employees, when this data should be removed, where this data should be stored, and possibly the anonymization of data to reduce privacy risks.

While Type 1 provides robust historical data, the downside is that it requires greater data governance practices. In contrast, Type 2 SCD has less (but not zero) privacy risk since only the current address is stored. Type 2 is simpler with less risk since old addresses are not retained, which reduces the risk of exposing outdated or sensitive information if there is a data breach. Nevertheless, inherent in the fact that personal information is still being collected and stored, there should still be adherence to Canadian privacy regulations on the handling, storage, access, control and removal of this information.

QUESTION 4

Review the AdventureWorks Schema. Highlight at least two differences between it and your ERD. Would you change anything in yours?

There are several differences between the Adventure Works schema and my bookstore ERD and the following highlights some of my observations between the two along with potential changes I would make to my bookstore db architecture:

1. Schema Segmentation

The AdventureWorks database is significantly more complex with multiple schemas that is organized according to different aspects of the business and its various supply chain entities, such as Sales, Human Resources, Production and Purchasing. In contrast, the bookstore ERD has all of its related tables into a

single, simple schema. This works well for a small business but would likely not scale effectively for larger operations with distinct business functions, such as supply chain functions, HR, Finance, Sales, etc. ***Potential Change****:* If the bookstore grows into a larger business with a need to split its business functions into separate entities, I could consider separating the bookstore db for better organization and scaling. For instance, a Sales schema would manage all customer-facing orders and sales, while a Purchasing schema would organize all inventory and publisher/vendor management and purchases.

2. Use of Bridge (Join) Tables for Many-to-Many Relationships

Another notable difference in the AdventureWorks ERD is their extensive use of bridge tables to handle many-to-many relationships. For example, the PersonCreditCard table bridges the relationship between Person and CreditCard, allowing a person to have multiple credit cards and a credit card to be linked to multiple persons. In contrast, the bookstore ERD does not include many-to-many relationships. For example, customers can only have a direct one-to-many relationship with orders with no direct support for scenarios with more complexity (e.g. one person orders a book as a gift to a second person).
Potential Change*:* If the bookstore grows, bridge tables could be used for future many-to-many relationships. For example, if one book order can be tied to multiple customers (e.g. in the case of gift orders), a CustomerOrder bridge table could be created.

3. Advanced Tracking of Data (Audit Tables and Status Codes)

AdventureWorks has status codes in several tables to track the current state of entities. For example, the SalesOrderHeader table contains a Status column that tracks the order's state (e.g. In Process, Shipped, Cancelled) along with timestamps for date creation and date modification (e.g. ModifiedDate). In contrast, the bookstore ERD is much more simplified with no tracking of "work-in-progress", and therefore, lacks tracking ability to determine how something has changed over time given that the current architecture has not status codes as in AdventureWorks.
Potential Change*:* Implementing more sophisticated **status codes** or **audit fields** would allow the bookstore to track not only the current state of an order but also changes over time (e.g. created_date, modified_date, status_code). These audit fields and status codes can provide valuable insights into the lifecycle of an order or an employee's status (e.g. history of an employee's promotions).

4. Temporal Data and Historical Tracking

Similar to the aforementioned point, AdventureWorks has been designed to track historical data, such as employee department history (EmployeeDepartmentHistory) or credit card usage (PersonCreditCard). These tables allow for tracking changes over time, which can be very powerful. On the other hand, my bookstore ERD this kind of historical tracking. For example, I do not track when an employee's position changes or a customer's address changes over time.
Potential Change*:* Adding history tables or using temporal columns (e.g., valid_from, valid_to) would allow the bookstore to keep track of changes, such as an employee moving from one position to another or a customer changing their address, which would enhance reporting and allow for better historical data analysis even though greater privacy implications and risk are involved.