CS4224/CS5424 Project

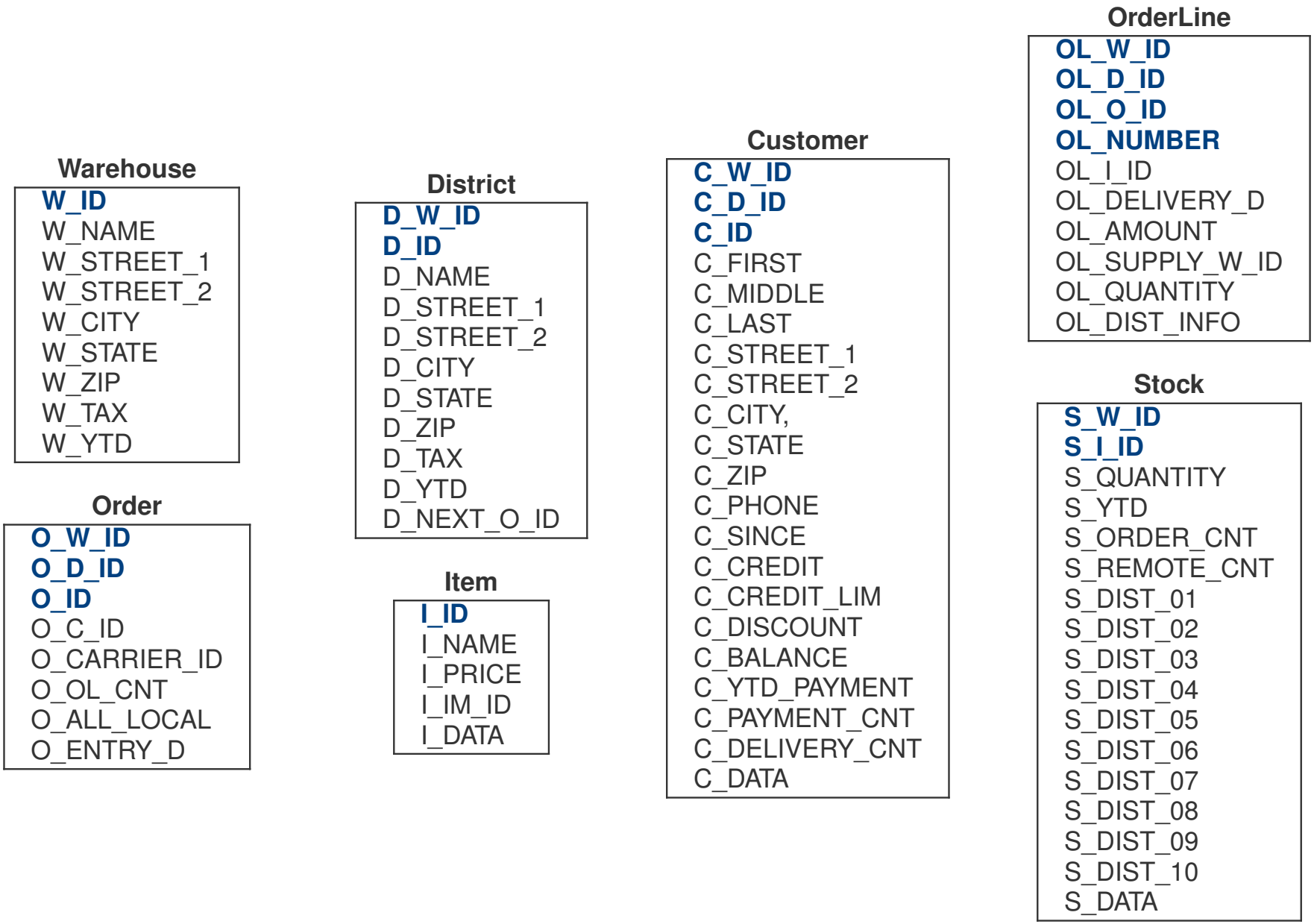
Objectives

- Provide students the opportunity to acquire practical experience with using distributed database systems for application development.
- Learning Tasks:
  - How to install a distributed database system on a cluster of machines
  - How to design a data model and implement transactions to support an application by taking into consideration of the application workload and features of the database system
  - How to benchmark the performance of an application
- Develop an application for a wholesale supplier using two different APIs of YugabyteDB:
  - YSQL: PostgreSQL-compatible API
  - YCQL: Cassandra-compatible API

Wholesale Supplier Application

- 5 entities (Warehouse, District, Customer, Order, Item) and
- 2 relationships (Order-Line, Stock)
  - Order-Line specifies the items contained in each customer's order
  - Stock specifies the availability information of items in each warehouse
- 10 warehouses
- Each warehouse covers 10 districts $\Longrightarrow$ 100 districts
- Each district serves 3000 customers $\Longrightarrow$ $30,000$ customers per warehouse
- 100,000 items per warehouse

Schema


Warehouse
W_ID
W_NAME
W_STREET_1
W_STREET_2
W_CITY
W_STATE
W_ZIP
W_TAX
W_YTD
District
D_W_ID
D_ID
D_NAME
D_STREET_1
D_STREET_2
D_CITY
D_STATE
D_ZIP
D_TAX
D_YTD
D_NEXT_O_ID
Customer
C_W_ID
C_D_ID
C_ID
C_FIRST
C_MIDDLE
C_LAST
C_STREET_1
C_STREET_2
C_CITY,
C_STATE
C_ZIP
C_PHONE
C_SINCE
C_CREDIT
C_CREDIT_LIM
C_DISCOUNT
C_BALANCE
C_YTD_PAYMENT
C_PAYMENT_CNT
C_DELIVERY_CNT
C_DATA
OrderLine
OL_W_ID
OL_D_ID
OL_O_ID
OL_NUMBER
OL_I_ID
OL_DELIVERY_D
OL_AMOUNT
OL_SUPPLY_W_ID
OL_QUANTITY
OL_DIST_INFO
Stock
S_W_ID
S_I_ID
S_QUANTITY
S_YTD
S_ORDER_CNT
S_REMOTE_CNT
S_DIST_01
S_DIST_02
S_DIST_03
S_DIST_04
S_DIST_05
S_DIST_06
S_DIST_07
S_DIST_08
S_DIST_09
S_DIST_10
S_DATA
Order
O_W_ID
O_D_ID
O_ID
O_C_ID
O_CARRIER_ID
O_OL_CNT
O_ALL_LOCAL
O_ENTRY_D
Item
I_ID
I_NAME
I_PRICE
I_IM_ID
I_DATA

Warehouse

Attribute	Meaning	Type
W_ID	Warehouse number	INT
W_NAME	Warehouse name	VARCHAR(10)
W_STREET_1	Warehouse address	VARCHAR(20)
W_STREET_2	Warehouse address	VARCHAR(20)
W_CITY	Warehouse address	VARCHAR(20)
W_STATE	Warehouse address	CHAR(2)
W_ZIP	Warehouse address	CHAR(9)
W_TAX	Warehouse sales tax rate	DECIMAL(4,4)
W_YTD	Year to date amount paid to warehouse	DECIMAL(12,2)

District

Attribute	Meaning	Type
D_W_ID	Warehouse number	INT
D_ID	District number	INT
D_NAME	District name	VARCHAR(10)
D_STREET_1	District address	VARCHAR(20)
D_STREET_2	District address	VARCHAR(20)
D_CITY	District address	VARCHAR(20)
D_STATE	District address	CHAR(2)
D_ZIP	District address	CHAR(9)
D_TAX	District sales tax rate	DECIMAL(4,4)
D_YTD	Year to date amount paid to district	DECIMAL(12,2)
D_NEXT_O_ID	Next available order number for district	INT

D_W_ID is a foreign key that refers to Warehouse table.

Customer

Attribute	Meaning	Type
C_W_ID	Warehouse number	INT
C_D_ID	District number	INT
C_ID	Customer number	INT
C_FIRST	Customer name	VARCHAR(16)
C_MIDDLE	Customer name	CHAR(2)
C_LAST	Customer name	VARCHAR(16)
C_STREET_1	Customer address	VARCHAR(20)
C_STREET_2	Customer address	VARCHAR(20)
C_CITY,	Customer address	VARCHAR(20)
C_STATE	Customer address	CHAR(2)
C_ZIP	Customer address	CHAR(9)
C_PHONE	Customer phone	CHAR(16)
C_SINCE	Date and time when entry was created	TIMESTAMP
C_CREDIT	Customer credit status	CHAR(2)
C_CREDIT_LIM	Customer credit limit	DECIMAL(12,2)
C_DISCOUNT	Customer discount rate	DECIMAL(4,4)
C_BALANCE	Balance of customer's outstanding payment	DECIMAL(12,2)
C_YTD_PAYMENT	Year to date payment by customer	FLOAT
C_PAYMENT_CNT	Number of payments made	INT
C_DELIVERY_CNT	Number of deliveries made to customer	INT
C_DATA	Miscellaneous data	VARCHAR(500)

(C_W_ID, C_D_ID) is a foreign key that refers to District table.

Order

Attribute	Meaning	Type
O_W_ID	Warehouse number	INT
O_D_ID	District number	INT
O_ID	Order number	INT
O_C_ID	Customer number	INT
O_CARRIER_ID	Identifier of carrier who delivered the order	INT
O_OL_CNT	Number of items ordered	DECIMAL(2,0)
O_ALL_LOCAL	Order status (whether order includes only home order-lines)	DECIMAL(1,0)
O_ENTRY_D	Order entry data and time	TIMESTAMP

(O_W_ID, O_D_ID, O_C_ID) is a foreign key that refers to Customer table. The range of O_CARRIER_IDis [1,10].

Item

Attribute	Meaning	Type
I_ID	Item identifier	INT
I_NAME	Item name	VARCHAR(24)
I_PRICE	Item price	DECIMAL(5,2)
I_IM_ID	Item image identifier	INT
I_DATA	Brand information	VARCHAR(50)

Order-Line

Attribute	Meaning	Type
OL_W_ID	Warehouse number	INT
OL_D_ID	District number	INT
OL_O_ID	Order number	INT
OL_NUMBER	Order-line number	INT
OL_I_ID	Item number	INT
OL_DELIVERY_D	Data and time of delivery	TIMESTAMP
OL_AMOUNT	Total price for ordered item	DECIMAL(6,2)
OL_SUPPLY_W_ID	Supplying warehouse number	INT
OL_QUANTITY	Quantity ordered	DECIMAL(2,0)
OL_DIST_INFO	Miscellaneous data	CHAR(24)

(OL_W_ID, OL_D_ID, OL_O_ID) is a foreign key that refers to Order table.
OL_I_ID is a foreign key that refers to Item table.
An order-line is classified as a home order-line if OL_SUPPLY_W_ID = OL_W_ID; otherwise, it it classified as a remote order-line. An order's O_ALL_LOCAL is set to *true* if and only if all its order-lines are home order-lines.

Stock

Attribute	Meaning	Type
S_W_ID	Warehouse number	INT
S_I_ID	Item number	INT
S_QUANTITY	Quantity in stock for item	DECIMAL(4,0)
S_YTD	Year to date total quantity ordered	DECIMAL(8,2)
S_ORDER_CNT	Number of orders	INT
S_REMOTE_CNT	Number of remote orders	INT
S_DIST_01	Information on district 1's stock	CHAR(24)
S_DIST_02	Information on district 2's stock	CHAR(24)
S_DIST_03	Information on district 3's stock	CHAR(24)
S_DIST_04	Information on district 4's stock	CHAR(24)
S_DIST_05	Information on district 5's stock	CHAR(24)
S_DIST_06	Information on district 6's stock	CHAR(24)
S_DIST_07	Information on district 7's stock	CHAR(24)
S_DIST_08	Information on district 8's stock	CHAR(24)
S_DIST_09	Information on district 9's stock	CHAR(24)
S_DIST_10	Information on district 10's stock	CHAR(24)
S_DATA	Miscellaneous data	VARCHAR(50)

S_I_ID is a foreign key that refers to Item table. S_W_ID is a foreign key that refers to Warehouse table.

Transaction Types

1. **New Order Transaction** processes a new customer order.
2. **Payment Transaction** processes a customer payment for an order.
3. **Delivery Transaction** processes the delivery of the oldest yet-to-be-delivered order for each of the 10 districts in a specified warehouse.
4. **Order-Status Transaction** queries the status of the last order of a specified customer.
5. **Stock-Level Transaction** checks the stock level of a specified number of last items sold at a warehouse district.
6. **Popular-Item Transaction** identifies the most popular items sold in each of a specified number of last orders at a specified warehouse district.
7. **Top-Balance Transaction** identifies the top-10 customers with the highest outstanding payment balance.
8. **Related-Customer Transaction** identifies the customers related to a specified customer.

Transaction Workload

Transaction Type	Frequency (%)
New-Order	40
Payment	20
Delivery	20
Order-Status	4
Stock-Level	4
Popular-Item	4
Top-Balance	6
Related-Customer	2

1. Number of items per new order: $[5, 20]$
2. Stock threshold T in Stock-level transactions: $[10, 20]$
3. Number L in Stock-level and Popular-item transactions: $[20, 50]$

Project Scope

For each YugabyteDB API (YSQL & YCQL), perform the following tasks to optimize the transaction throughput for the workload:

1. Design a data model
2. Implement a function for each of the 8 transaction types
3. Install & configure YugabyteDB on a cluster of 5 servers
4. Benchmark the performance of the implementation for the workload
   - Implement a main driver program that simulates a client executing transactions on the database
     - ★ Program reads its inputs from stdin until EOF
     - ★ Each input specifies an instance of one of the 8 transaction types
     - ★ For each transaction read, program invokes the appropropriate transaction function
   - Benchmark the performance of the implementation

Project Deliverables

- Due on October 7 (Friday, 11:59pm):
  - One-page progress report
- Due on November 4 (Friday, 11:59pm):
  - Code
  - Project report