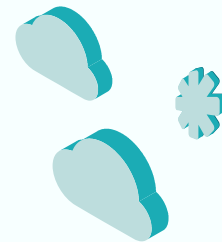


# Ellise Ski Shop Database

Eddie Deschappelles, Laura Yan,  
Lauren Hanley, Liren Zhou

December 14th, 2023  
Group B





# Table of contents

**01**

**Project Framing**

**02**

**Entity-Relation  
Diagram**

**03**

**Relational Schema  
Mapping**

**04**

**Data Sources**

**05**

**Queries**

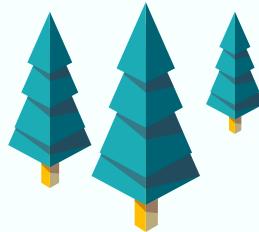
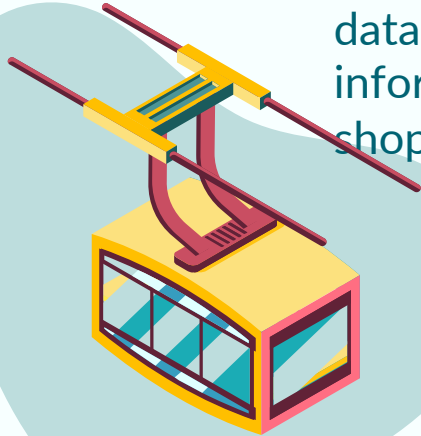
**06**

**Visualizations**

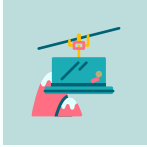


# About our company

Ellise began as a small ski shop in 2018. With the rapid expansion and growing popularity of Ellise over the past few years, there is a pressing need for an ER database system to comprehensively document all information and transaction activities for the Ellise ski shop chain.

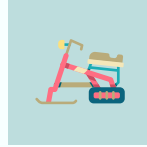


# What are the issues?



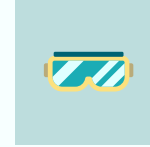
## Ineffective

- We are proposing a new database system that streamlines all information about the Ellise ski shops and appropriately organizes details about employees, shops, products, customers, sales, and rentals.



## Inaccurate

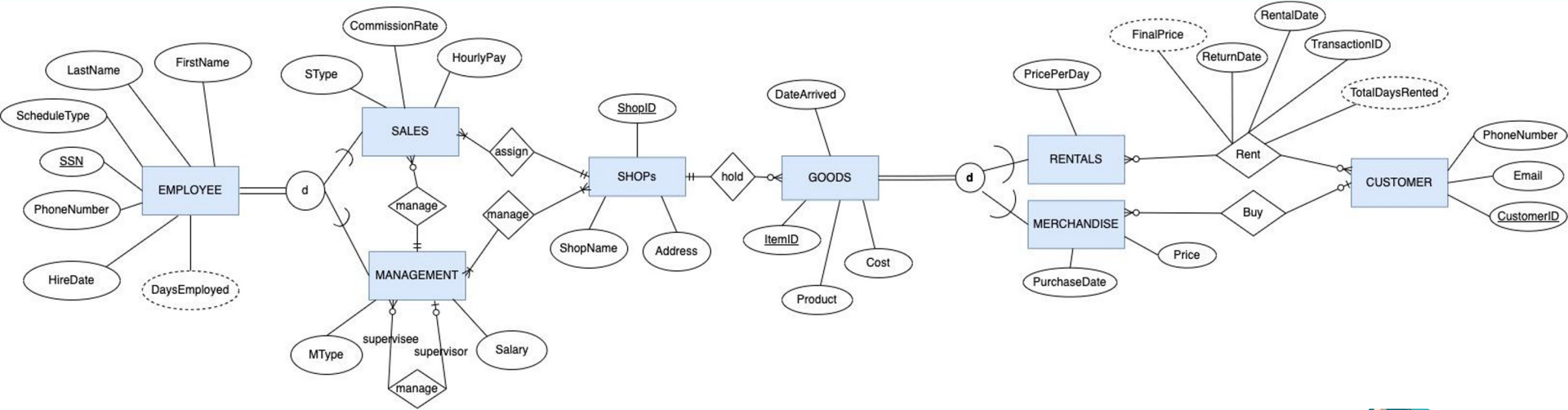
- Ellise's current database system does not accurately track Ellise's employees. Our ER system differentiates between sales and management positions.



## Disorganized

- Our ER system focuses on ensuring that goods and customers are accurately accounted for and tracked.

# Entity-Relation Diagram



# Relational Schema Mapping

- EMPLOYEE( SSN, ScheduleType, FirstName, LastName, PhoneNumber, HireDate)
- MANAGEMENT( SSN(FK), Salary, MType, Supervisor[FK] )
- SALES ( SSN(FK), CommissionRate, SType, HourlyPay, ShopID(FK), Supervisor[FK] )
- Manage ( SSN(FK), shopID(FK) )
- SHOPS ( ShopID, Address, ShopName )
- GOODS ( ItemID, Cost, Product, DateArrived, shopID[FK])
- RENTALS ( ItemID(FK), PricePerDay )
- Rent ( ItemID(FK), CustomerID(FK), RentalDate, ReturnDate, TransactionID)
- MERCHANDISE ( ItemID(FK), CustomerID(FK), PurchaseDate, Price)
- CUSTOMER ( CustomerID, PhoneNumber, Email)



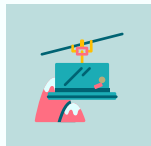
# Data Sources

- For our data sources, we primarily used random data from an online platform.
- Employee, Goods, Rentals, Merchandise & Customer Tables: We randomly generated the data for these tables with data from Mockaroo.com.
- Shops Table: This data was collected from ski town locations in Colorado.



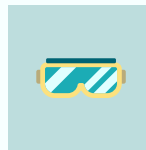


# Query Report #1



## Purpose

- The purpose is to retrieve the top-selling products based on the total quantity sold and total income.
- This financial information helps the Ellise team understand customer buying patterns, which informs inventory, pricing, and promotions.



## SQL Commands

```
SELECT Product,  
COUNT(Merchandise.ItemId) AS  
TotalQuantitySold, SUM(Price) AS  
TotalIncome  
FROM Merchandise  
JOIN Goods ON Merchandise.ItemId =  
Goods.ItemID  
WHERE CustomerID <> "  
GROUP BY Product  
ORDER BY TotalQuantitySold DESC;
```









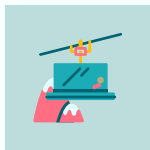
# Query Report #1 Output



Product ▲	TotalQuantitySold ▲	TotalIncome ▲
Ski Pant	3	1022.94
Snowboard	2	650.12
Skis	2	1032.26
Ski Poles	2	654.90
Goggles	2	779.97

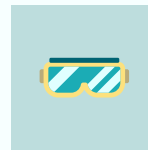


# Query Report #2



## Purpose

- The purpose of this query is to determine which merchandise items arrived more than 60 days ago and haven't been sold.
- This is helpful for marketing efforts because it allows Ellise to push a 10% discount promotion that will get rid of merchandise sitting on the shelves.



## SQL Commands



```
SELECT *  
FROM Goods  
WHERE DATEDIFF(CURDATE(),  
STR_TO_DATE(DateArrived,  
'%m/%d/%Y')) > 60  
AND ItemID NOT IN (SELECT ItemID  
FROM Merchandise);
```



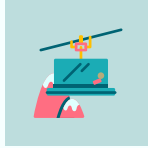
# Query Report #2 Output



ItemID ▲	Cost ▲	Product ▲	DateArrived ▲	ShopID ▲
18	\$142.09	Ski Poles	10/3/2023	2
20	\$93.18	Ski Pant	9/3/2023	3
21	\$52.32	Skis	9/17/2023	3

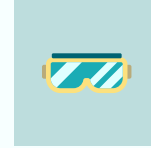


# Query Report #3



## Purpose

- The purpose is to retrieve the average commission rate and hourly pay for each sales associate type.
- This is helpful for understanding employee performance and compensation.



## SQL Commands

```
SELECT SType,  
AVG(CommissionRate) AS  
AvgCommissionRate,  
AVG(HourlyPay) AS  
AvgHourlyPay  
FROM Sales  
GROUP BY SType;
```



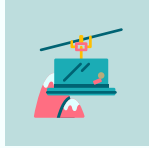
# Query Report #3 Output



SType ▲	AvgCommissionRate ▲	AvgHourlyPay ▲
Junior Sales Associate	0.02	15.350000
Sales Associate	0.08	19.466667
Senior Sales Associate	0.08	22.565000



# Query Report #4



## Purpose

- For employee management, the purpose of this query is to find out which employees have worked the longest at Ellise.
- Based on this information, Ellise can create a promotion to reward the loyalty of these employees.



## SQL Commands



```
SELECT SSN, FirstName,  
        LastName,  
        DATEDIFF(CURDATE(),  
        STR_TO_DATE(HireDate,  
        '%m/%d/%Y')) as  
        DaysEmployed  
FROM Employee  
ORDER BY DaysEmployed  
DESC;
```



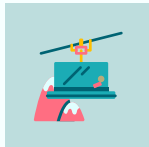
# Query Report #4 Output



SSN ▲	FirstName ▲	LastName ▲	DaysEmployed ▲
229-44-0992	Mylo	Biles	1713
786-65-3346	Welbie	Jaegar	1491
894-87-7081	Ceil	Nucci	1468
478-06-5004	Alidia	Prydden	1461
269-19-5271	Osborn	Tschirschky	1436
129-60-6774	Bogey	Gruszecki	924
754-74-8535	Arlene	Bilson	832
337-74-1030	Ariel	Howchin	547
492-59-1994	Lou	Hubbocks	506
525-92-8636	Genevra	Mullins	26



# Query Report #5



## Purpose

- The purpose of this query is to display the total number of day that goods have been rented, some of which have been rented multiple times, which we had to account for in our SQL commands.
- This query also shows the total revenue from the rented goods based on the price per day and total days rented.
- Based on this information, Ellise can determine which rented goods are the most popular and have contributed the most to the revenue.



## SQL Commands

```
SELECT Rentals.*,  
       IFNULL(T1.TotalDaysRented, 0) AS TotalDaysRented,  
       FORMAT(IFNULL(T1.TotalDaysRented, 0) *  
(REPLACE(Rentals.PricePerDay, '$', '') + 0), 2) AS TotalRevenue  
FROM Rentals  
LEFT JOIN (  
  SELECT  
    ItemID,  
    SUM(DATEDIFF(  
      STR_TO_DATE(ReturnDate, '%m/%d/%Y'),  
      STR_TO_DATE(RentalDate, '%m/%d/%Y')  
    )) AS TotalDaysRented  
  FROM Rent  
  GROUP BY ItemID  
) AS T1 ON Rentals.ItemID = T1.ItemID;
```






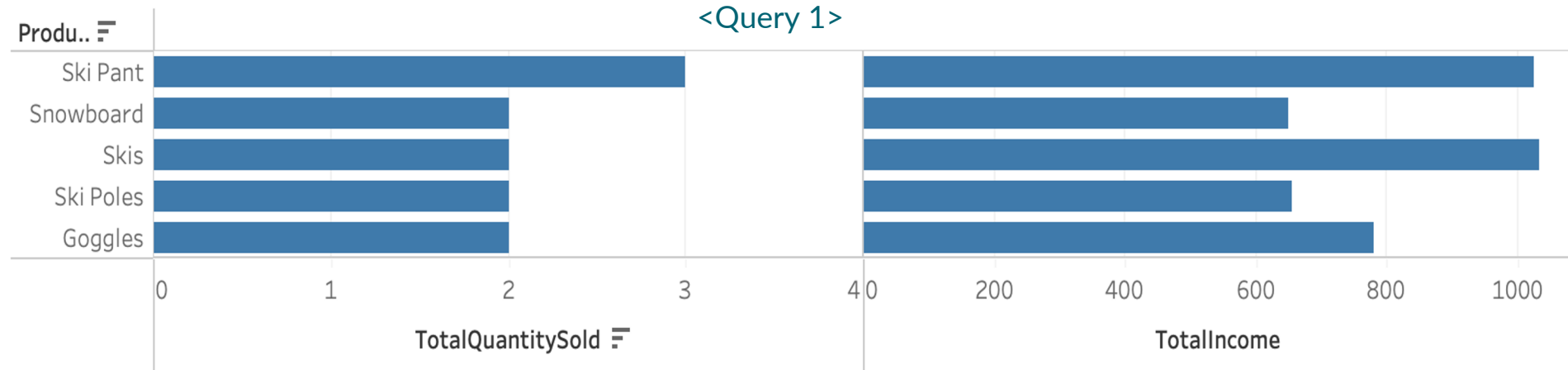
# Query Report #5 Output



ItemId ▲	PricePerDay ▲	TotalDaysRented ▲	TotalRevenue ▲
16	\$77.22	7	540.54
17	\$48.15	6	288.90
18	\$62.65	10	626.50
19	\$40.35	22	887.70
20	\$14.65	0	0.00
21	\$12.40	13	161.20
22	\$37.42	13	486.46
23	\$23.24	0	0.00
24	\$83.95	17	1,427.15
25	\$72.44	3	217.32

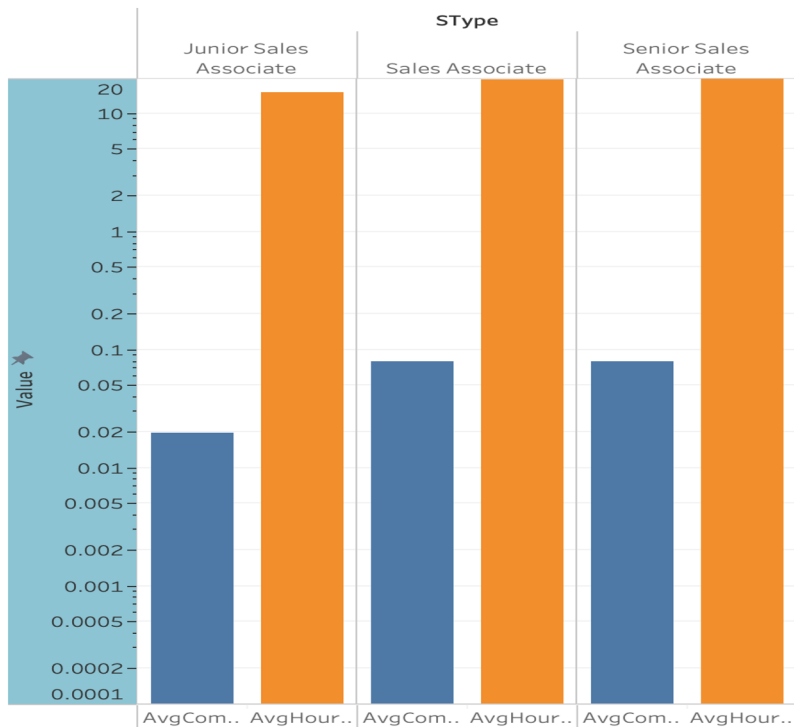


# Visualizations

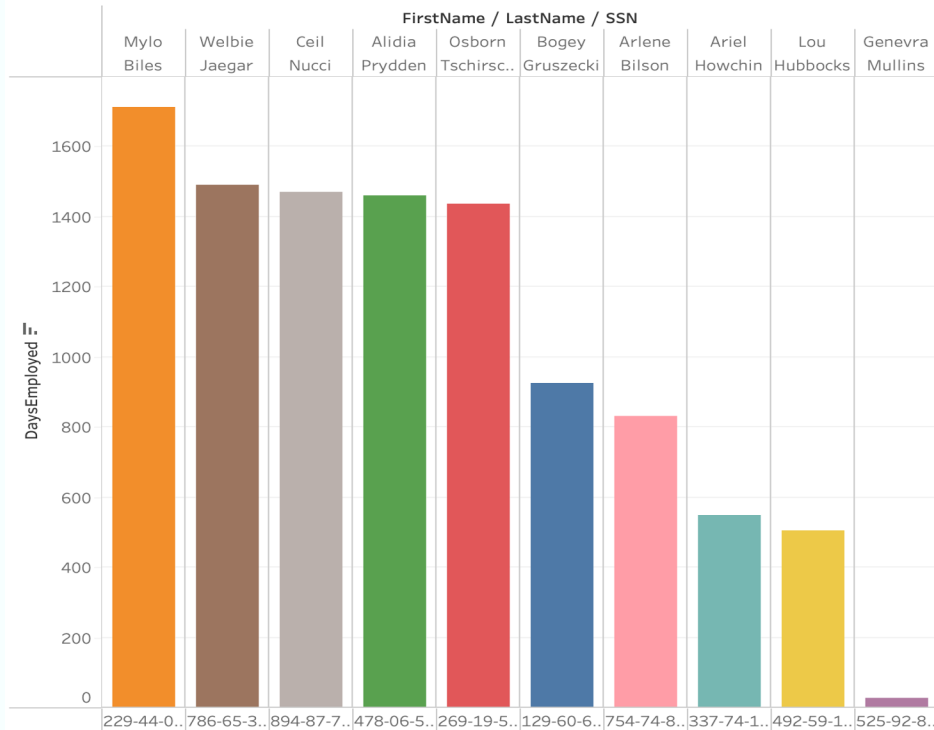


# Visualizations

<Query 3>

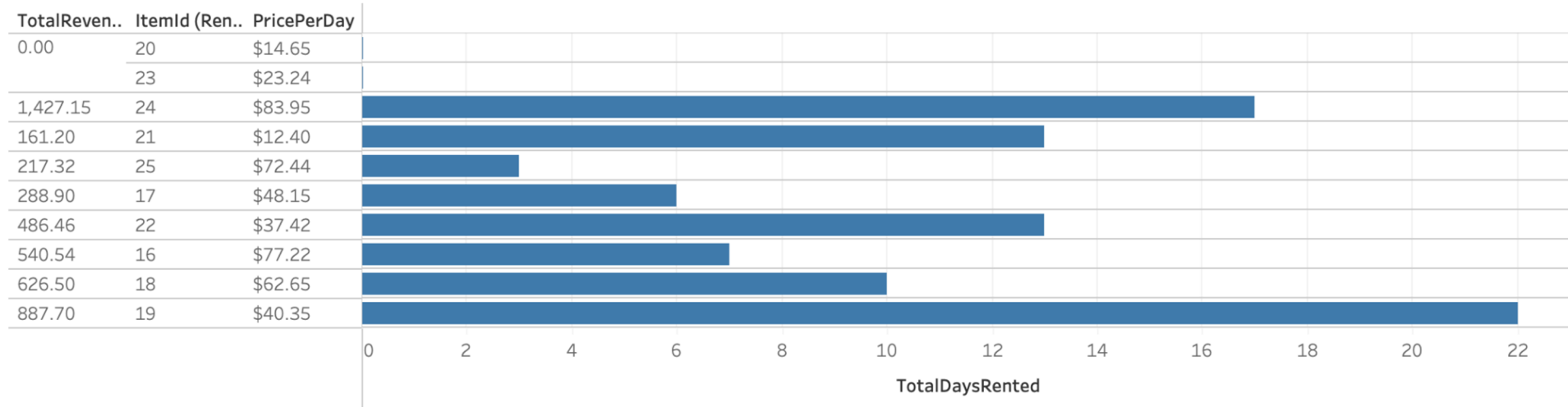


<Query 4>



# Visualizations

<Query 5>



# Lessons Learned & Future Directions

- Through working on this project, we learned more about the practical application of ER diagrams, relational schema, and SQL queries.
- One particular obstacle that we encountered was that SQL interpreted the currency values in our database as a string. We were able to adjust this issue by replacing the \$ and then converting the value to a decimal.
- One direction for the future is to plan for database scalability to accommodate potential future growth in the number of shops, merchandise, and customers of Ellise.
- Another way to improve upon this database is to incorporate a feedback mechanism, so that both customers and employees can review their experience with Ellise.



# Thanks!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**

**Please keep this slide for attribution**

