STARBUCKS:
Mobile Ordering Process

The Inner Joins:
Eshna Reza, Sneha Batchu, Sony Kumari,
Alex Soek, Billy Marin

Agenda

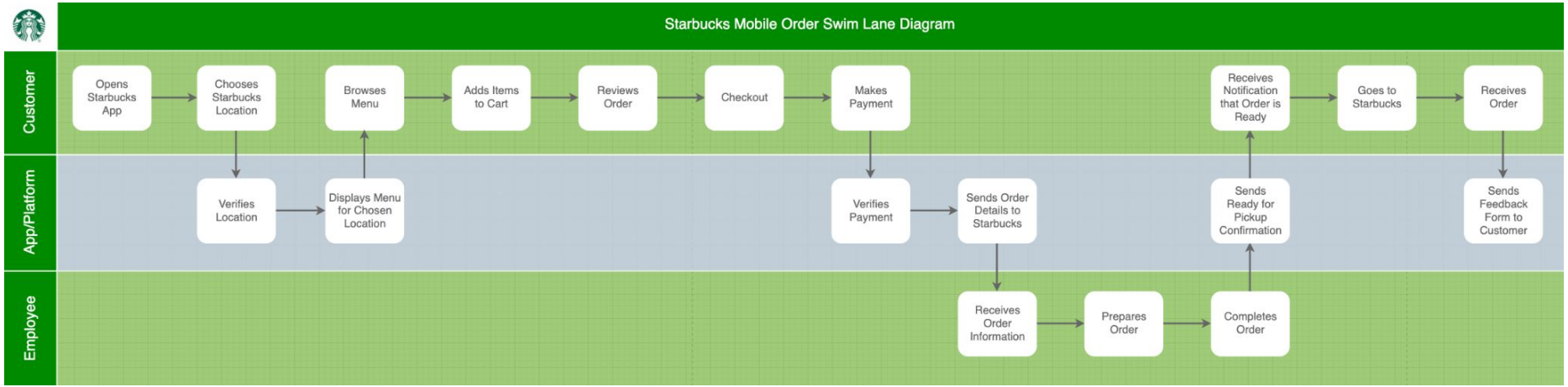
- Business Scenario and Process
- ERD and Tables
- SQL Queries
- Advanced SQL Queries
- Visualizations
- Conclusion

Business Scenario

A customer is getting ready for work and decides she is craving her favorite coffee and maybe a small breakfast, but she's in a hurry. She opens her Starbucks mobile app and decides to pick something up on her way to the office. She selects the Starbucks location closest to her, browses the menu, and selects a Venti Blonde latte and a blueberry muffin. After she adds her payment information, she chooses to have her order ready immediately, and then heads out the door at 8:15am.

Starbucks accepts the order and sends the customer a notification her order will be ready by 8:30am. A barista prepares the order and sets it in the pick up area. The customer receives an update her order is ready just as she enters the parking lot at 8:29am. She goes in to the counter and finds her items, thanks the barista, and leaves at 8:33am. After she leaves, she received a notification to provide feedback on her experience.

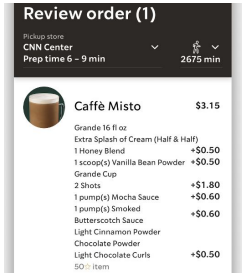
Business Process Diagram



Tables

customer

customer_order

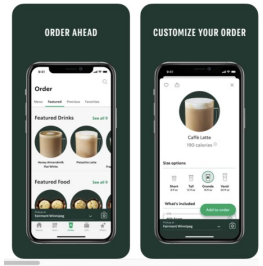


customer_order_detail

employee

item_master

loyalty_rewards_info

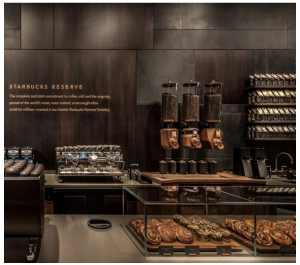


menu

order_feedback

payment

store



store_inventory

supplier

Table Creation

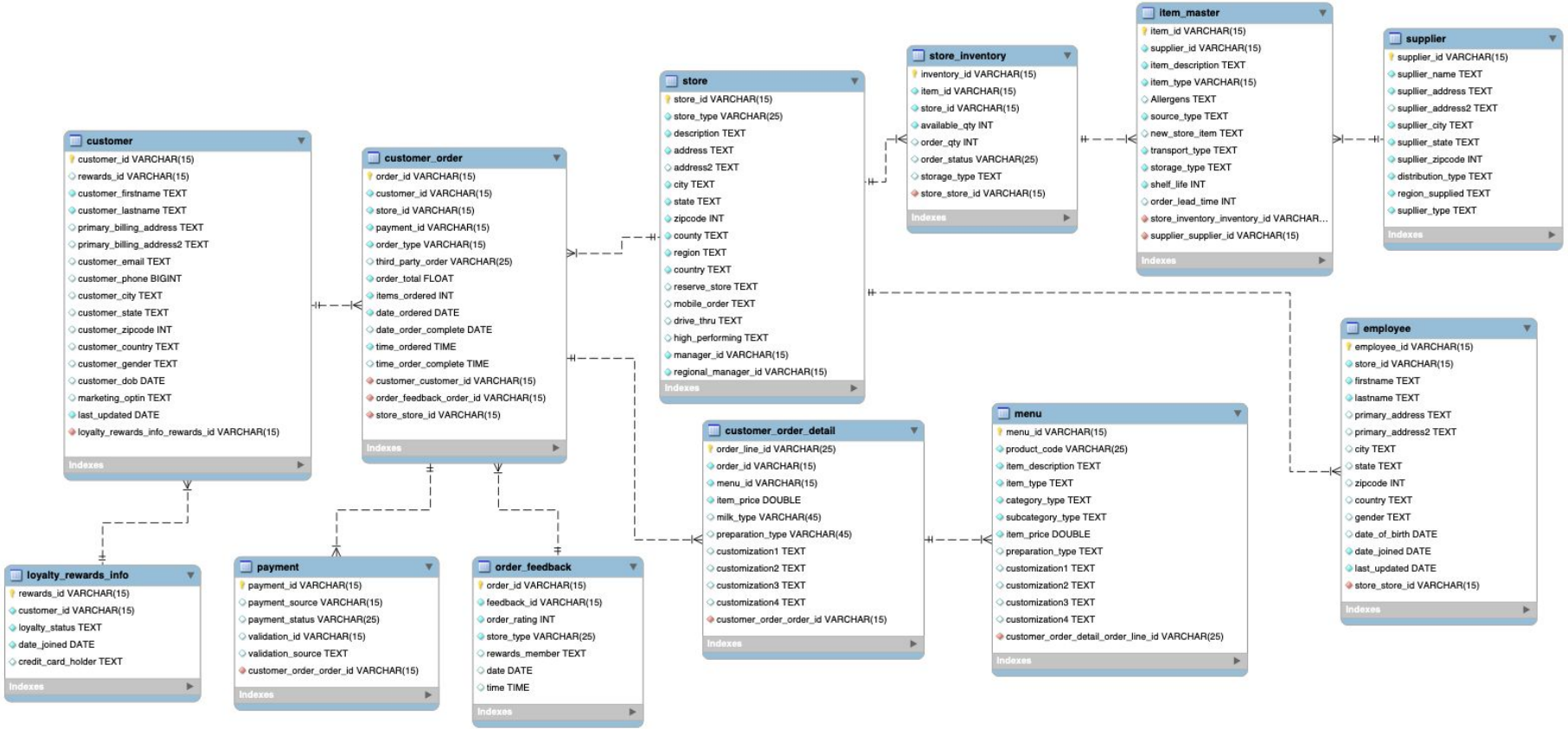
Used data upload tool in MySQL Workbench and used model to sync tables and constraints

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_

ALTER TABLE `Starbucks`.`customer`
ADD COLUMN `loyalty_rewards_info_rewards_id` VARCHAR(15) NOT NULL AFTER `last_updated`,
ADD INDEX `fk_customer_loyalty_rewards_info1_idx` (`loyalty_rewards_info_rewards_id` ASC) VISIBLE;
;

ALTER TABLE `Starbucks`.`customer_order`
ADD COLUMN `customer_customer_id` VARCHAR(15) NOT NULL AFTER `time_order_complete`,
ADD COLUMN `order_feedback_order_id` VARCHAR(15) NOT NULL AFTER `customer_customer_id`,
ADD COLUMN `store_store_id` VARCHAR(15) NOT NULL AFTER `order_feedback_order_id`,
ADD INDEX `fk_customer_order_customer_idx` (`customer_customer_id` ASC) VISIBLE,
ADD INDEX `fk_customer_order_order_feedback1_idx` (`order_feedback_order_id` ASC) VISIBLE,
ADD INDEX `fk_customer_order_store1_idx` (`store_store_id` ASC) VISIBLE;
;

ALTER TABLE `Starbucks`.`customer_order_detail`
ADD COLUMN `customer_order_order_id` VARCHAR(15) NOT NULL AFTER `customization4`,
ADD INDEX `fk_customer_order_detail_customer_order1_idx` (`customer_order_order_id` ASC) VISIBLE;
;

ALTER TABLE `Starbucks`.`employee`
ADD COLUMN `store_store_id` VARCHAR(15) NOT NULL AFTER `last_updated`,
ADD INDEX `fk_employee_store1_idx` (`store_store_id` ASC) VISIBLE;
;

ALTER TABLE `Starbucks`.`item_master`
ADD COLUMN `store_inventory_inventory_id` VARCHAR(15) NOT NULL AFTER `order_lead_time`,
ADD COLUMN `supplier_supplier_id` VARCHAR(15) NOT NULL AFTER `store_inventory_inventory_id`,
ADD INDEX `fk_item_master_store_inventory1_idx` (`store_inventory_inventory_id` ASC) VISIBLE,
ADD INDEX `fk_item_master_supplier1_idx` (`supplier_supplier_id` ASC) VISIBLE;
;

ALTER TABLE `Starbucks`.`menu`
ADD COLUMN `customer_order_detail_order_line_id` VARCHAR(25) NOT NULL AFTER `customization4`,
ADD INDEX `fk_menu_customer_order_detail1_idx` (`customer_order_detail_order_line_id` ASC) VISIBLE;
;
```

```
ALTER TABLE `Starbucks`.`payment`
ADD COLUMN `customer_order_order_id` VARCHAR(15) NOT NULL AFTER `validation_source`,
ADD INDEX `fk_payment_customer_order1_idx` (`customer_order_order_id` ASC) VISIBLE;
;

ALTER TABLE `Starbucks`.`store_inventory`
ADD COLUMN `store_store_id` VARCHAR(15) NOT NULL AFTER `storage_type`,
ADD INDEX `fk_store_inventory_store1_idx` (`store_store_id` ASC) VISIBLE;
;

ALTER TABLE `Starbucks`.`customer`
ADD CONSTRAINT `fk_customer_loyalty_rewards_info1`
  FOREIGN KEY (`loyalty_rewards_info_rewards_id`)
  REFERENCES `Starbucks`.`loyalty_rewards_info` (`rewards_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `Starbucks`.`customer_order`
ADD CONSTRAINT `fk_customer_order_customer`
  FOREIGN KEY (`customer_customer_id`)
  REFERENCES `Starbucks`.`customer` (`customer_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
ADD CONSTRAINT `fk_customer_order_order_feedback1`
  FOREIGN KEY (`order_feedback_order_id`)
  REFERENCES `Starbucks`.`order_feedback` (`order_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
ADD CONSTRAINT `fk_customer_order_store1`
  FOREIGN KEY (`store_store_id`)
  REFERENCES `Starbucks`.`store` (`store_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `Starbucks`.`customer_order_detail`
ADD CONSTRAINT `fk_customer_order_detail_customer_order1`
  FOREIGN KEY (`customer_order_order_id`)
  REFERENCES `Starbucks`.`customer_order` (`order_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;
```

```
ALTER TABLE `Starbucks`.`employee`
ADD CONSTRAINT `fk_employee_store1`
  FOREIGN KEY (`store_store_id`)
  REFERENCES `Starbucks`.`store` (`store_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `Starbucks`.`item_master`
ADD CONSTRAINT `fk_item_master_store_inventory1`
  FOREIGN KEY (`store_inventory_inventory_id`)
  REFERENCES `Starbucks`.`store_inventory` (`inventory_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE,
ADD CONSTRAINT `fk_item_master_supplier1`
  FOREIGN KEY (`supplier_supplier_id`)
  REFERENCES `Starbucks`.`supplier` (`supplier_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `Starbucks`.`menu`
ADD CONSTRAINT `fk_menu_customer_order_detail1`
  FOREIGN KEY (`customer_order_detail_order_line_id`)
  REFERENCES `Starbucks`.`customer_order_detail` (`order_line_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `Starbucks`.`payment`
ADD CONSTRAINT `fk_payment_customer_order1`
  FOREIGN KEY (`customer_order_order_id`)
  REFERENCES `Starbucks`.`customer_order` (`order_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;

ALTER TABLE `Starbucks`.`store_inventory`
ADD CONSTRAINT `fk_store_inventory_store1`
  FOREIGN KEY (`store_store_id`)
  REFERENCES `Starbucks`.`store` (`store_id`)
  ON DELETE CASCADE
  ON UPDATE CASCADE;
```

Entity Relationship Diagram (ERD)



SQL Queries

What is the most expensive order?

```sql
#What is the most expensive mobile order (by city)?
SELECT s.city AS Store_City, max(o.order_total) AS Max_Order_Total
FROM customer_order o, store s
WHERE s.store_id = o.store_id
GROUP BY s.city;
```

100% 53:72

Result Grid Filter Rows: Search Export:

Store_City	Max_Order_Total
Santa Clara	21
San Jose	7

What is the least expensive item on the menu?

```sql
#What is the least expensive item on the menu?
SELECT item_description, item_type, item_price
FROM menu
WHERE item_price = (SELECT min(item_price) FROM menu);
```

100% 47:99

Result Grid Filter Rows: Search Export:

item_description	item_type	item_price
brown_sugar_syrup	add_ons	0.5
caramel_syrup	add_ons	0.5
chestnut_praline_syrup	add_ons	0.5
cinnamon_dolce_syrup	add_ons	0.5
hazelnut_syrup	add_ons	0.5
irish_cream_syrup	add_ons	0.5
peppermint_syrup	add_ons	0.5
raspberry_syrup	add_ons	0.5
sugar_cookie_syrup	add_ons	0.5
toffee_nut_syrup	add_ons	0.5
vanilla_syrup	add_ons	0.5
sugar_free_vanilla_sy...	add_ons	0.5
caramel_drizzle	add_ons	0.5
mocha_sauce	add_ons	0.5
dark_caramel_sauce	add_ons	0.5
pistachio_sauce	add_ons	0.5

SQL Queries with Group By

What is the average wait time of customers?

What is the peak time of the day for ordering?

```
74  #What is the average wait time of customers?
75  SELECT order_type, COUNT(order_type) as order_type_count,
76  AVG(TIMESTAMPDIFF(MINUTE, time_ordered, time_order_complete)) AS avg_time_in_mins
77  FROM customer_order
78  GROUP BY customer_order.order_type
79  ORDER BY avg_time_in_mins DESC;
```

100% 45:74

Result Grid Filter Rows: Search Export:

order_type	order_type_count	avg_time_in_mins
Third Party	2	21.0000
Mobile	5	13.6000
In-Person	2	7.5000

```
58  #What is the peak time of the day for ordering?
59  SELECT COUNT(order_id) as Orders,
60  CONCAT(Hour(time_ordered)) as Hour_Ordered
61  FROM customer_order
62  GROUP BY Hour_Ordered
63  ORDER BY Orders desc;
64
```

100% 48:58

Result Grid Filter Rows: Search Export:

Orders	Hour_Ordered
3	8
2	9
1	15
1	11
1	10
1	13

SQL Queries with Group By

Which items were ordered the most from the menu?

What payment types have been used and how much was the order total of each?

```
57  #Which items were ordered the most from the menu?
58  SELECT menu.menu_id, menu.item_description AS Item_Name,
59  COUNT(customer_order_detail.order_line_id) AS Number_of_Customer_Orders
60  FROM Starbucks.menu
61  INNER JOIN Starbucks.customer_order_detail
62  ON menu.menu_id = customer_order_detail.menu_id
63  GROUP BY menu.menu_id
64  ORDER BY COUNT(customer_order_detail.order_line_id) DESC LIMIT 5;
```

100% 50:57

Result Grid | Filter Rows: Search | Export:

menu_id	Item_Name	Number_of_Customer_Orders
m065	impossible_bfast_swich	4
m066	bacon_egg_cheese_swich	2
m069	blueberry_muffin	2
m064	pink_drink_venti	2
m042	caramel_frap_grande	1

```
93  #What payment types have been used and how much was the order total of each?
94  SELECT customer_order.payment_id, payment.payment_source, customer_order.order_total
95  FROM customer_order
96  LEFT JOIN payment ON customer_order.payment_id = payment.payment_id
97  ORDER BY customer_order.order_total desc;
```

100% 77:93

Result Grid | Filter Rows: Search | Export:

payment_id	payment_sour...	order_total
p05a	paypal	21
p04i	reward_points	18
p02d	credit_card	15
p05d	paypal	14
p02a	credit_card	13.25
p04a	reward_points	11
p01	cash	10.5
p03i	gift_card	7
p03a	gift_card	5.5

SQL Queries with Group By

Which zipcode has the most number of online mobile orders?

How many Starbucks are there in Santa Clara and San Jose?

```
48    #Which zipcode has the most number of online mobile orders?
49    SELECT zipcode , sum(order_count) AS total_online_mobile_order from (
50    SELECT customer_order.store_id, count(customer_order.store_id) AS order_count, zipcode
51    FROM customer_order
52    INNER JOIN store ON store.store_id = customer_order.store_id
53    WHERE customer_order.order_type = "Mobile"
54    GROUP BY customer_order.store_id) as T1
55    GROUP BY zipcode order by total_online_mobile_order desc LIMIT 1;
```

100% 60:48

Result Grid | Filter Rows: Search | Export:

zipcode	total_online_mobile_order
95051	2

```
78    #How many Starbucks are there in Santa Clara and San Jose ?
79    SELECT city, COUNT(store_id) AS store_count
80    FROM store
81    WHERE city in ('Santa Clara', 'San Jose')
82    GROUP BY city;
```

100% 60:78

Result Grid | Filter Rows: Search | Export:

city	store_count
San Jose	9
Santa Clara	13

SQL Queries with Joins

How many people have loyalty program?

```sql
#How many people have loyalty program?
SELECT l.loyalty_status AS Rewards_Status, COUNT(DISTINCT l.customer_id) AS Rewards_Members_Count
FROM customer c
INNER JOIN loyalty_rewards_info l ON l.customer_id = c.customer_id
GROUP BY l.loyalty_status;
```

100% 27:34

Result Grid | Filter Rows: Search | Export:

Rewards_Status	Rewards_Members_Count
Gold	6
Green	5
Silver	4

```sql
#How often do loyalty program customers order beverages?
SELECT loyalty_rewards_info.loyalty_status as loyalty_status,
COUNT(DISTINCT customer_order.customer_id) AS loyalty_members_count,
COUNT(DISTINCT customer_order_detail.menu_id) AS distinct_beverages_ordered
FROM loyalty_rewards_info
JOIN customer_order
ON customer_order.customer_id = loyalty_rewards_info.customer_id
JOIN customer_order_detail
ON customer_order.order_id = customer_order_detail.order_id
WHERE customer_order_detail.menu_id < 'm065'
GROUP BY loyalty_rewards_info.loyalty_status;
```

100% 57:36

Result Grid | Filter Rows: Search | Export:

loyalty_status	loyalty_members_count	distinct_beverages_ordered
Gold	4	6
Green	3	2
Silver	1	1

How often do loyalty program customers order beverages?

Advanced SQL Queries

Stored Procedure

Using a stored procedure that returns peak ordering day

```
DELIMITER $$

CREATE PROCEDURE GetPeakDay()
BEGIN
    SELECT COUNT(order_id) AS Orders,
    ELT(DAYOFWEEK(GetPeakDayate_ordered),
    'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat') as Day_of_Week
    FROM customer_order
    GROUP BY Day_of_Week;
END$$
DELIMITER ;
call Starbucks.GetPeakDay();
```

100% 17:18

Result Grid | Filter Rows: Search | Export:

Orders	Day_of_Week
2	Sun
2	Mon
1	Tue
1	Wed
1	Thu
1	Fri
1	Sat

Advanced SQL Queries

Subqueries

Using a subquery to return if hot or cold drink type that is ordered more often, and how many times the drink type was ordered

```sql
SELECT customer_order_detail.preparation_type AS Prep_Type,
COUNT(customer_order_detail.order_line_id) AS Count_prep_type
FROM customer_order_detail
INNER JOIN menu
ON menu.menu_id = customer_order_detail.menu_id
WHERE menu.item_type = 'drink'
GROUP BY Prep_Type
HAVING COUNT(customer_order_detail.order_line_id) = (
    SELECT MAX(mycount)
        FROM (
            SELECT customer_order_detail.preparation_type,
            COUNT(customer_order_detail.order_line_id) AS mycount
            FROM customer_order_detail
            INNER JOIN menu
            ON menu.menu_id = customer_order_detail.menu_id
            WHERE menu.item_type = 'drink'
            GROUP BY customer_order_detail.preparation_type
            ) AS results
        )
    ;
```

100% 29:33

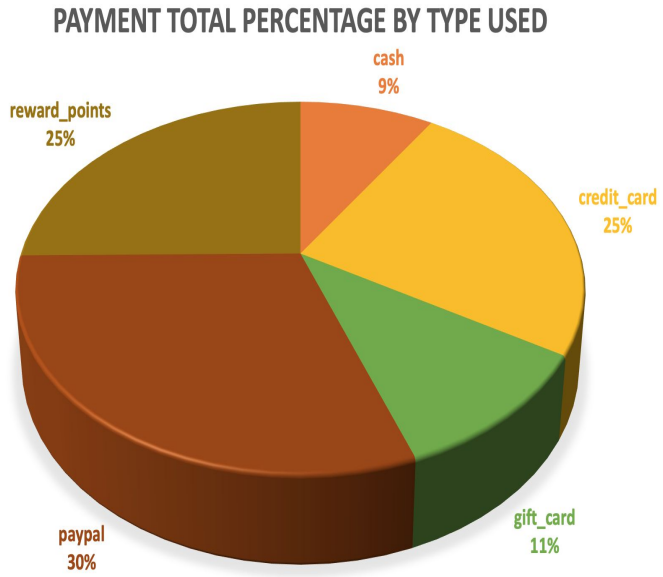
Result Grid | Filter Rows: Search | Export:

Prep_Type	Count_prep_type
Cold	6

Visualization - By Payment Type

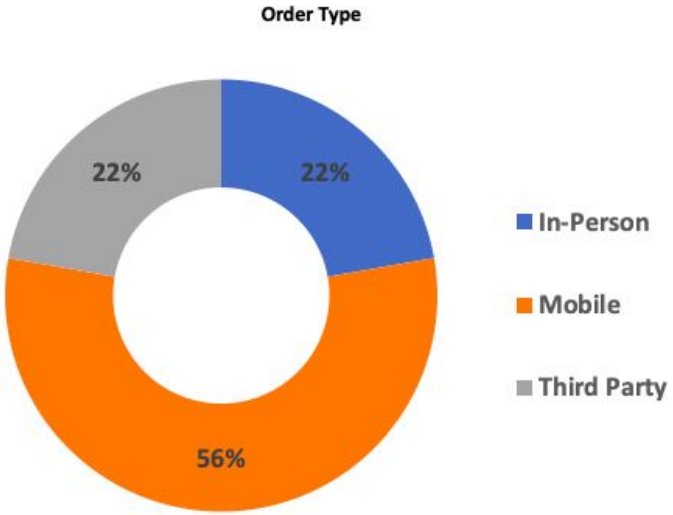
```
SELECT  customer_order.customer_id, customer_order.payment_id, customer_order.items_ordered,
        customer_order.order_total, payment.payment_source
FROM customer_order
LEFT JOIN payment ON customer_order.payment_id = payment.payment_id
ORDER BY payment.payment_source;
```

customer_id	payment_id	items_ordered	order_total	payment_source
9	p01	2	10.5	cash
4	p02a	2	13.25	credit_card
13	p02d	3	15	credit_card
1	p03a	1	5.5	gift_card
6	p03i	2	7	gift_card
11	p05a	4	21	paypal
3	p05d	2	14	paypal
7	p04a	2	11	reward_points
8	p04i	3	18	reward_points



Visualization using Excel

Visualization - By Order Type

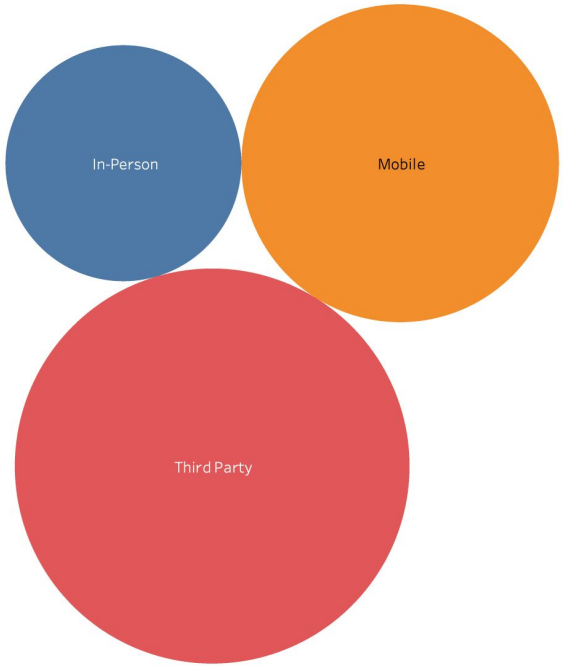


order_id	customer_id	store_id	payment_id	order_type
101	9	sc1	p01	In-Person
102	4	sc6	p02a	Mobile
103	13	sc13	p02d	Mobile
104	1	sc4	p03a	Mobile
105	6	sc14	p03i	In-Person
106	7	sc18	p04a	Mobile
107	8	sc1	p04i	Mobile
108	11	sc20	p05a	Third Party
109	3	sc3	p05d	Third Party

Visualization using Excel

Visualization - By Wait Time

By Wait Time

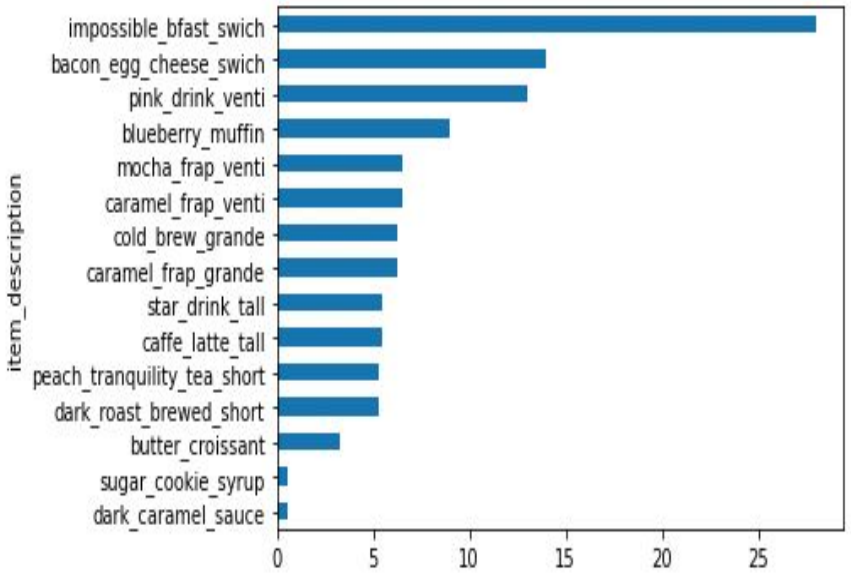


order_type	order_type_count	avg_time_in_mins
Third Party	2	21.0000
Mobile	5	13.6000
In-Person	2	7.5000

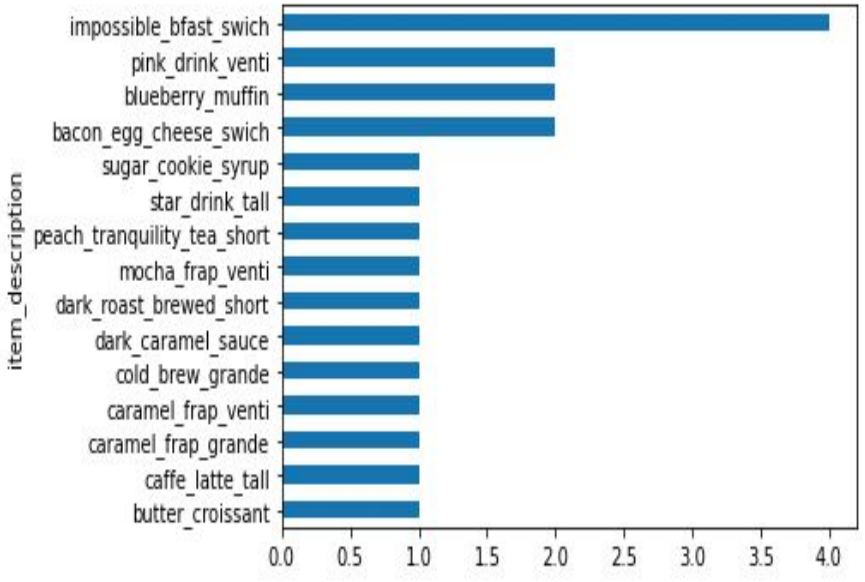
Visualization using Tableau

Visualization - Totals of items orders

By sum of costs



By count of items



Visualization using Python

Conclusions

- What did you learn from this project
  - How to construct simple and more complex SQL Queries
  - How to create business process diagrams
  - How to design and build a database structure
- What problems did you encounter during the project
  - Making sure database is connected
    - Primary keys, foreign keys/constraints
  - Creating mock data
- How did you solve these problems?
  - Group discussions
  - Trial and error
  - Testing, testing, testing
- If you were to do this project again, what would you do differently
  - Plan out business questions and then define data around them
  - Build out data scenarios first then build tables
  - Talk to real life users or create a sample order
- Such a simple process such as creating a mobile order has lots of behind the scenes tasks

Thank you!

Questions?