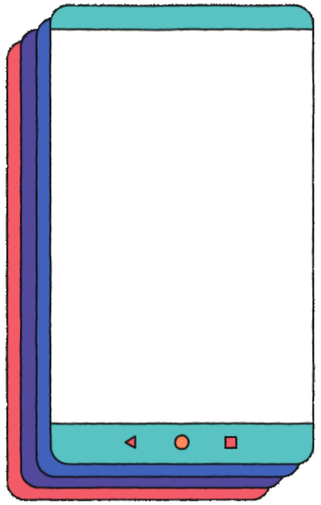


Database - 2606

High Distinction Assignment

Digi - Smart Retailer

Student Details

Student Name - Vishal Poojary

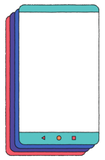
Student ID - 13739863

Autumn Semester

Content

- Scenario & Purpose
- ERD - Entity Relationship Diagram
- SQL - Queries and Output
- Data Integrity

Digi - Smart Retailer



- **Smartphone**
- Accessories
- Support services

Major brands as supplier

Purpose

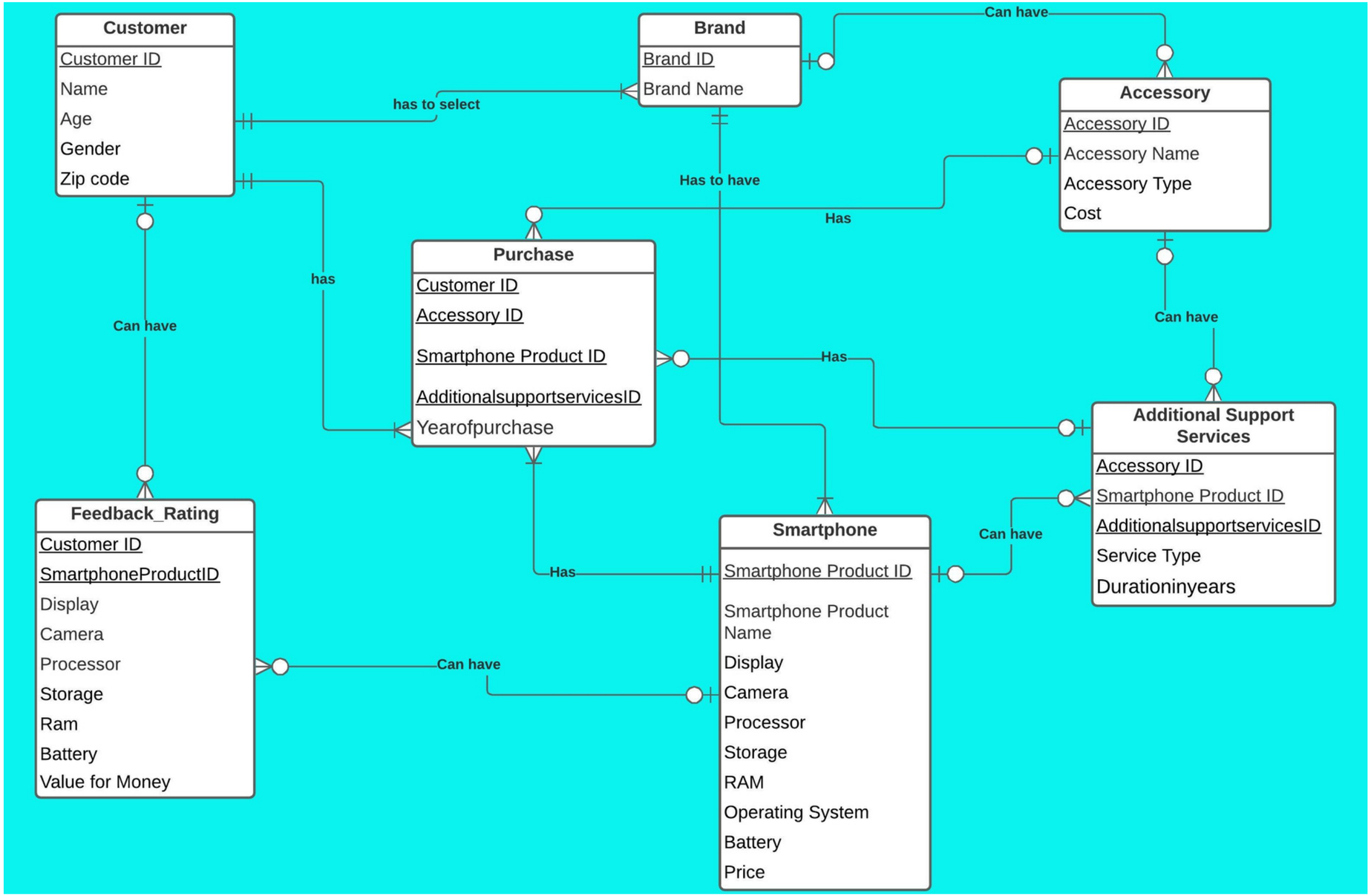
To store and monitor data of day to day business activities

Customer ⟶ **Product & Services** ⟶ **Purchase** ⟶ **Feedback**

↓

ERD

Digi - Smart Retailer ERD



One to Many relationship between entities

- One Brand can have many smartphones (1:M)
- Brand ID becomes a foreign key in the smartphone table

Brand

Brand ID	Customer ID	Brand Name
001	1001	Google
002	1001	Apple

Smartphone

Smartphoneproduct ID	Smartphone product name	Display	Camera	Processor	Storage	Ram	OS	Battery	Price	BrandID
2001	Pixel 2	1920*1080	12 MP	Octa - Core	64 GB	4 GB	Android	3000 MAH	250	001
2004	Pixel 2 XL	1440*2880	12 MP	Octa - Core	64 GB	4 GB	Android	3500 MAH	500	001
2002	Iphone XR	1792*828	12 MP	Hexa - Core	64 GB	3 GB	IOS	2942 MAH	700	002
2005	Iphone 11	1792*828	12 MP	Hexa - Core	64 GB	4 GB	IOS	3000 MAH	800	002

Many to Many relationships between Entities

- Many smartphones can be compatible with many accessories and vise versa

- A weak entity (additionalsupportservices) is created as a bridge to handle many to many relationship

Accessory

Accessory ID	Accessory Name	AccessoryType	Price	Brand ID
3001	Pixel Buds	Earphone	50	001
3002	Pixel Leather	Case	100	001
3007	Airpod	Earphone	70	002

AdditionalSupportServices

Additionalsupportservices ID	Accessory ID	Smartphone Product ID	Service Type	Duration in years
4001	3001	2009	Platinum	2
4004	3002	2001	Silver	1
4007	3007	2008	Platinum	3
4008	3007	2011	Gold	2

Smartphone

Smartphoneproduct ID	Smartphone product name	Display	Camera	Processor	Storage	Ram	OS	Battery	Price	BrandID
2001	Pixel 2	1920*1080	12 MP	Octa - Core	64 GB	4 GB	Android	3000 MAH	250	001
2008	Iphone 12 Pro Max	1440*2880	12 MP	Octa - Core	128 GB	4 GB	IOS	3500 MAH	1500	002
2009	Pixel 4 XL	1440*2880	12 MP	Octa-Core	64 GB	6 GB	Android	3500 MAH	1000	001
2011	Iphone 12	1920*1080	12 MP	Hexa - Core	64 GB	4 GB	IOS	3000 MAH	800	002

SQL Query & Output

- Simple Query

Smartphones offered by Google

Note - BrandID 01 = 'Google'

Query - Select smartphoneID, smartphoneproductname from Smartphone where brandID = 01;

```
postgres=# Select smartphoneproductID, smartphoneproductname from Smartphone where brandID = 01;
 smartphoneproductid | smartphoneproductname
---------------------+-----------------------
 2001 | Pixel 2
 2004 | Pixel 2XL
 2006 | Pixel 4a
 2009 | Pixel 4XL
 2012 | Pixel 4
 2015 | Pixel 5
(6 rows)
```

SQL Query & Output

- Natural join & Groupby Query

Number of smartphones offered by brands order by highest to lowest ?

Query - Select brandname, Count(smartphoneproductID) as "Number of smartphones" from Brand natural join smartphone group by brandname order by Count(smartphoneproductID) DESC;

```
postgres=# Select brandname, Count(smartphoneproductID) as "Numberofsmartphones" from Brand natural join smartphone group by brandname order
by Count(smartphoneproductID) DESC;
 brandname | Numberofsmartphones
-----------+---------------------
 Google    |                   6
 Apple     |                   5
 Oneplus   |                   4
 Huawei    |                   2
 Realme    |                   2
 Nokia     |                   2
 HTC       |                   2
 LG        |                   2
 Motorola  |                   2
 Xiaomi    |                   2
 Vivo      |                   2
 Oppo      |                   2
 Samsung   |                   2
 Asus      |                   2
 Sony      |                   2
(15 rows)
```

SQL Query & Output

- Cross-Product Query

Which brands have high switching cost compared to other brands?

Note: It is implied brands which offer more than just smartphones (i.e additional support services and accessories) will incur high switching cost to the Consumers

Query - Select DISTINCT Brand.brandID, Brand.brandname from Smartphone, Brand, Accessory, Additionalsupportservices where Brand.brandID = Smartphone.brandID and Brand.brandID = Accessory.brandID order by Brand.brandID;

```
postgres=# Select  DISTINCT Brand.brandID, Brand.brandname from Smartphone, Brand, Accessory, Additionalsupportservices where Brand.brandID = Smartphone.brandID and Brand.brandID = Accessory.brandID order by Brand.brandID;
 brandid | brandname
---------+-----------
       1 | Google
       2 |  Apple
       3 | Oneplus
      10 |  Asus
      12 |  LG
(5 rows)
```

SQL Query & Output

- Sub Query & Group By followed by 'having'

Smartphones that consumers consider low value for their money?

Query - Select smartphoneproductname, Sum(valueformoney) as "LowValueforMoney" from smartphone, Feedback_Rating where Smartphone.smartphoneproductID = Feedback_Rating.smartphoneproductID group by smartphoneproductname having Sum(valueformoney) <= (Select AVG(valueformoney) from Feedback_Rating);

```
postgres=# Select smartphoneproductname, Sum(valueformoney) as "LowValueforMoney" from smartphone, Feedback_Rating where Smartphone.smartphon
eproductID = Feedback_Rating.smartphoneproductID group by smartphoneproductname having Sum(valueformoney) <= (Select AVG(valueformoney) from
Feedback_Rating);
 smartphoneproductname | LowValueforMoney
-----------------------+------------------
  X20                  |                3
  G8 ThinkQ            |                3
  G100                 |                3
  Xperia 1             |                3
  ROG Phone III        |                3
 Vivo V20              |                2
 Realme 8 Pro          |                3
 Oppo A54              |                2
 Galaxy S20            |                3
Xiaomi Mi 11 Ultra     |                3
HTC U12+               |                3
Huawei P30 Pro         |                3
(12 rows)
```

Data Integrity

Business requirement

1. An efficient way to delete suppliers from the database
2. Safeguard customer data
3. Values of feedback need to be out of 5 and cannot be NULL
4. Types of accessory sold by the retailer are limited to
    a. Case
    b. Charger
    c. Earphone
    d. Smartwatch

Referential Integrity

- Prevents entry of duplicate data
- Can construct a logical relationship between entities

Req 1 - Delete supplier from database

"ON DELETE CASCADE"

When a record in the primary table (Brand) is deleted all the corresponding data in the referenced tables (Smartphone, accessory) will also get deleted automatically

Brand

brandid	customerid	brandname
1	1001	Google
2	1001	Apple
3	1002	Oneplus
4	1003	Samsung
5	1003	Huawei
6	1004	Xiaomi
7	1005	Oppo
8	1005	Vivo
9	1006	Sony
10	1007	Asus
11	1007	Realme
12	1008	LG
13	1009	Nokia
14	1009	Motorola
15	1010	HTC

Smartphone

smartphoneproductid	smartphoneproductname	display	camera	processor	storage	ram	operatingsystem	battery	price	brandid
2001	Pixel 2	1920*1080	12 Mega Pixel	Octa-Core	64 GB	4 GB	Android	3000 MAH	250	1
2002	Iphone XR	1792*828	12 Mega Pixel	Hexa-Core	64 GB	3 GB	IOS	2942 MAH	700	2
2003	OnePlus 8T	1920*1080	48 Mega Pixel	Octa-Core	128 GB	8 GB	Android	4500 MAH	500	3

Accessory

accessoryid	accessoryname	accessorytype	price	brandid
3001	Pixel Buds	Earphone	50	1
3002	Pixel Leather	Case	100	1
3003	Google Pixel Stand	Charger	60	1
3004	McSafe	Charger	80	2
3005	MiniClear	Case	150	2
3006	Max Leather	Case	200	2
3007	Airpod	Earphone	70	2
3008	OnePlus Buds	Earphone	95	3

Referential Integrity

Req 2 - Safeguard consumer data

"ON DELETE RESTRICT"

Consumer record cannot be deleted from the primary table (Customer) if any of the associated record exist in child table (Brand)

Customer

customerid	name	age	gender	zipcode
1001	Brendon	26	M	400071
1002	Sam	30	M	400018
1003	Samyuktha	24	F	400020
1004	Divya	30	F	400022
1005	James	32	M	400024
1006	Emma	34	F	400026
1007	Kate	36	F	400028
1008	Andrew	38	M	400028

Brand

brandid	customerid	brandname
1	1001	Google
2	1001	Apple
3	1002	Oneplus
4	1003	Samsung
5	1003	Huawei
6	1004	Xiaomi
7	1005	Oppo
8	1005	Vivo

Domain Integrity

- Ensures data is verified at the point the data is being added to the database
- Database will not accept data unless it meets the domain requirement

Req 3 - Values of feedback need to be out of 5 and cannot be NULL

```
CONSTRAINT di_table_rating_display CHECK
        ((display <= 5) AND (display >= 1)),

CONSTRAINT di_table_rating_camera CHECK
        ((camera <= 5) AND (camera >= 1)),

CONSTRAINT di_table_rating_processor CHECK
        ((processor <= 5) AND (processor >= 1)),

CONSTRAINT di_table_rating_storage CHECK
        ((storage <= 5) AND (storage >= 1)),

CONSTRAINT di_table_rating_ram CHECK
        ((ram <= 5) AND (ram >= 1)),

CONSTRAINT di_table_rating_battery CHECK
        ((battery <= 5) AND (battery >= 1)),

CONSTRAINT di_table_rating_valueformoney CHECK
        ((valueformoney <= 5) AND (valueformoney >= 1))
        );
```

Domain Integrity

- **Req 4** - Types of accessory sold by the retailer are limited to

  - Case
  - Charger
  - Earphone
  - Smartwatch

  CONSTRAINT di_table_accessory CHECK (accessorytype IN (
  'Case',
  'Charger',
  'Earphone',
  'Smartwatch'))

Thank you!