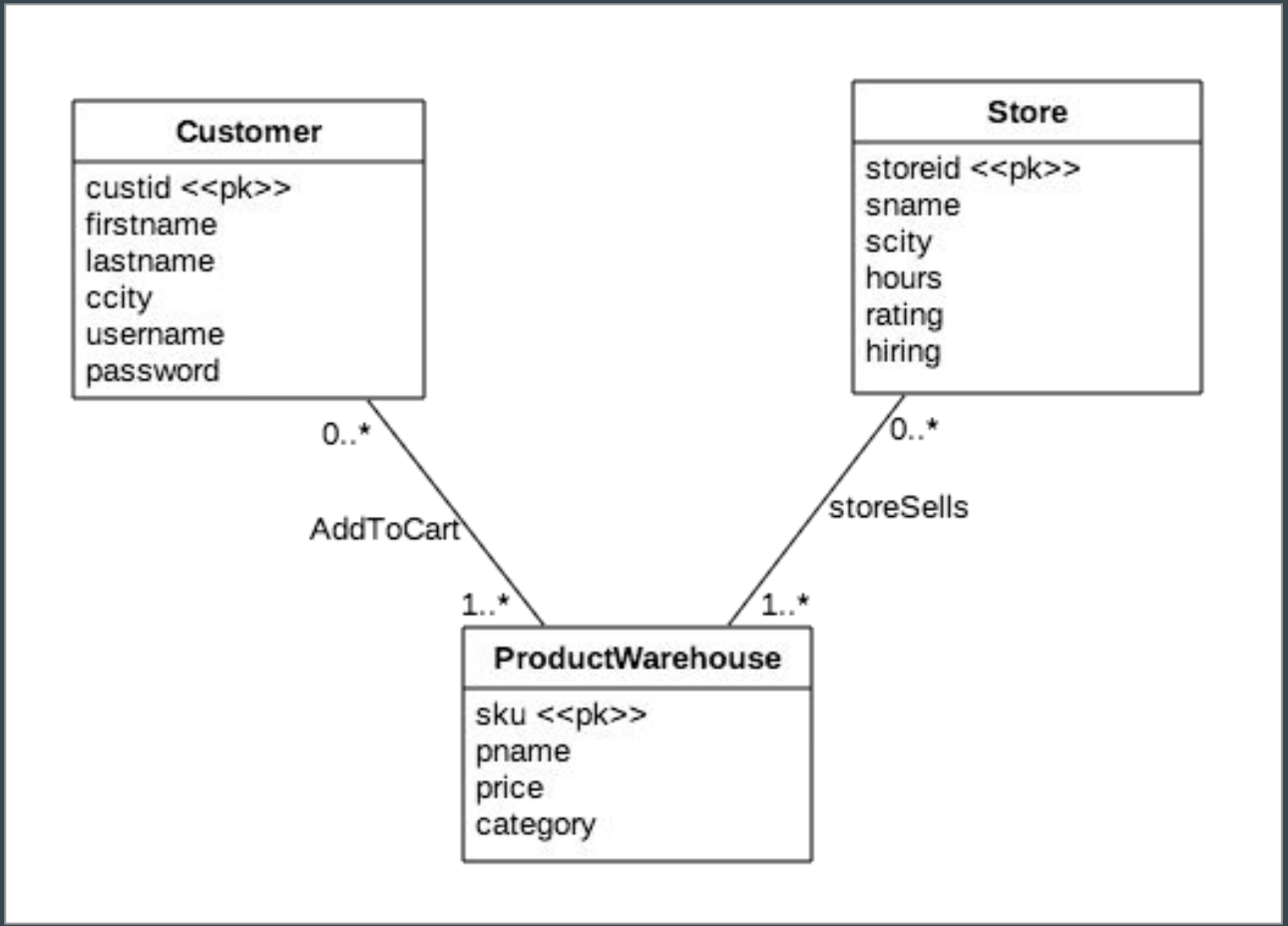
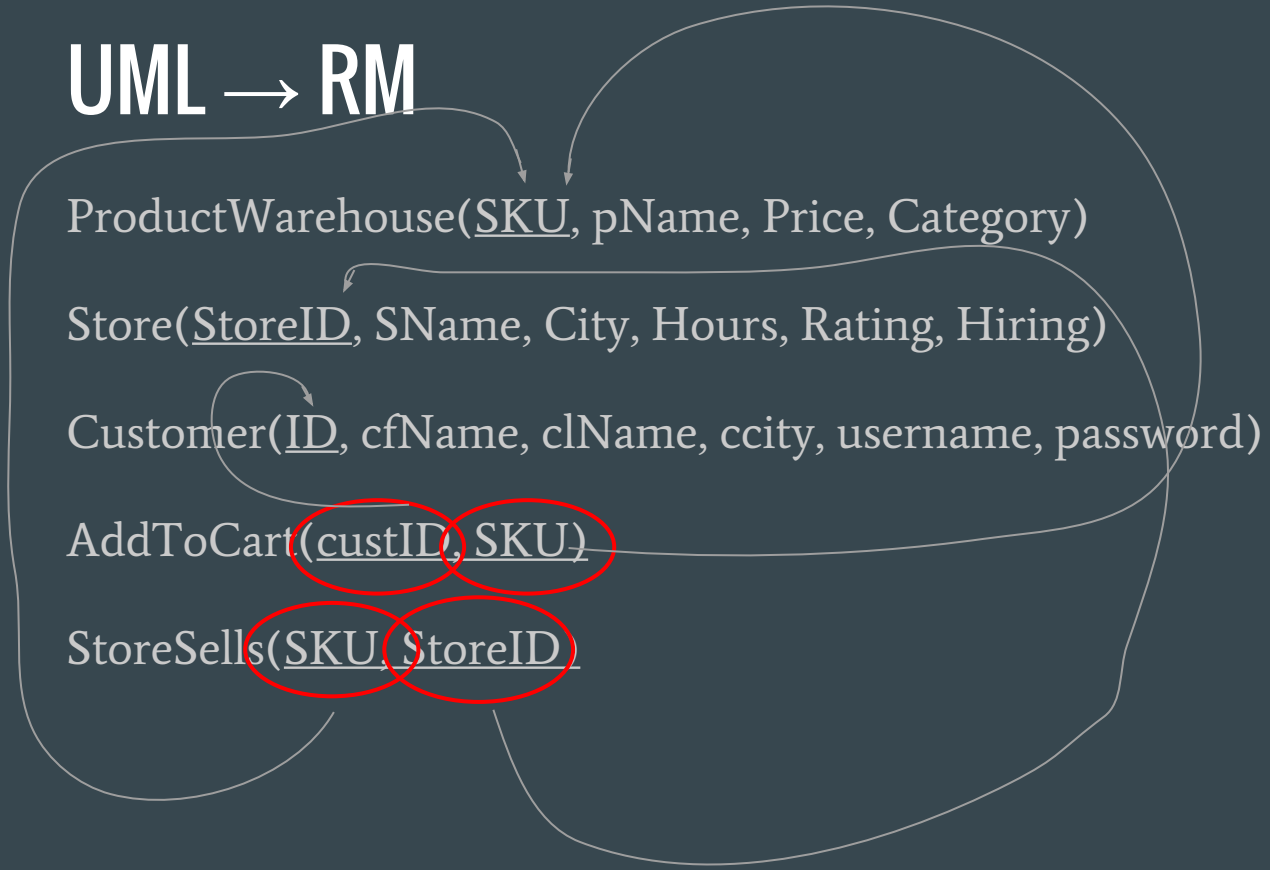
The Online Grocery App

Grayson Holt
Kosi Uzodinma
Molly Orr
Monica Korfas
Wesley Nguyen

UML


Customer
custid <<pk>>
firstname
lastname
ccity
username
password
Store
storeid <<pk>>
sname
scity
hours
rating
hiring
0..*
AddToCart
1..*
0..*
storeSells
1..*
ProductWarehouse
sku <<pk>>
pname
price
category

UML → RM



ProductWarehouse(SKU, pName, Price, Category)

Store(StoreID, SName, City, Hours, Rating, Hiring)

Customer(ID, cfName, clName, ccity, username, password)

AddToCart(custID, SKU)

StoreSells(SKU, StoreID)

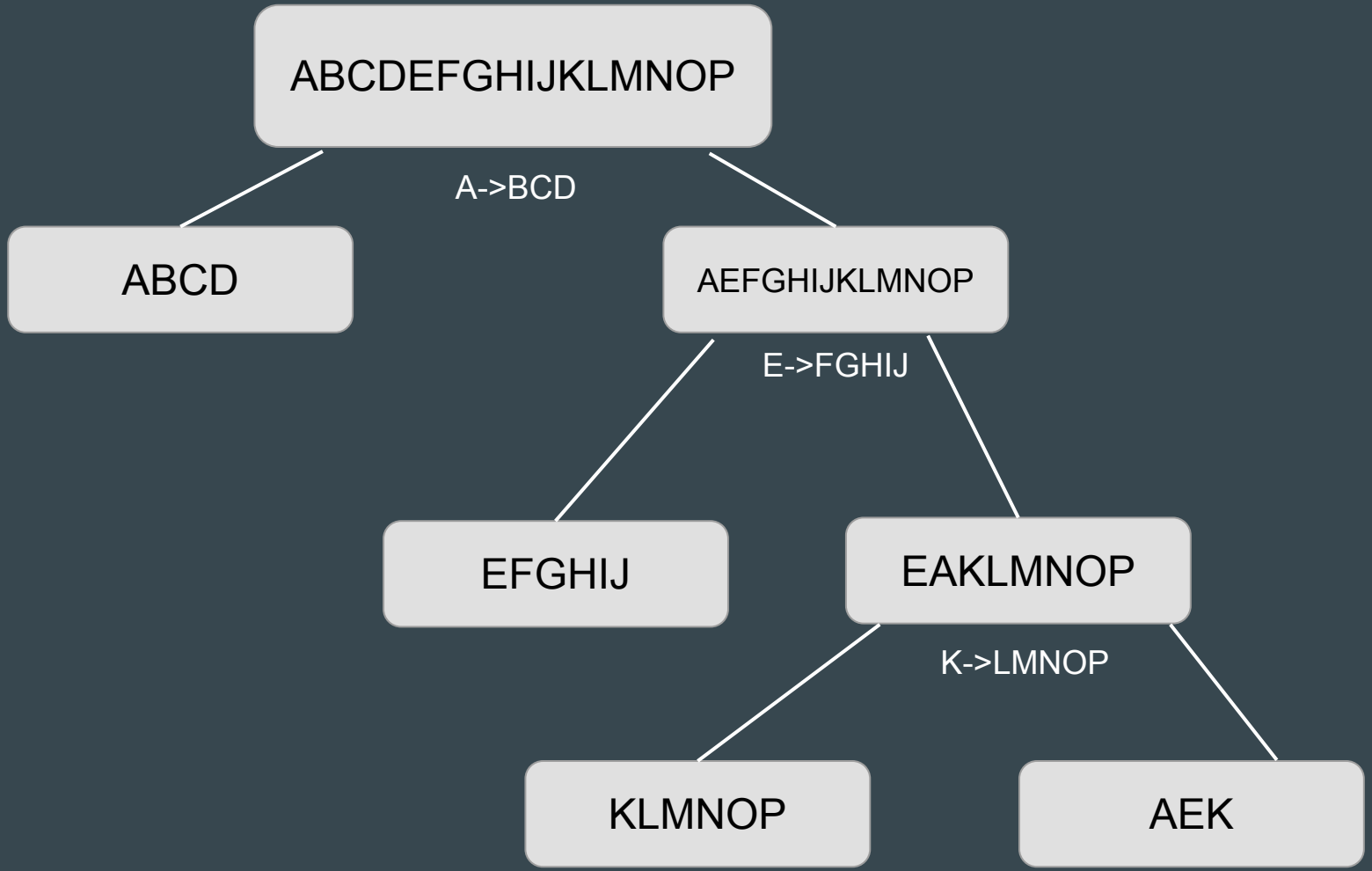
BCNF

A = SKU
B = pName
C = Price
D = Category
E = StoreID
F = SName
G = SCity
H = Hours
I = Rating
J = Hiring
K = custID
L = cfName
M = clName
N = ccity
O = username
P = password

R= ABCDEFGHIJKLMNOP

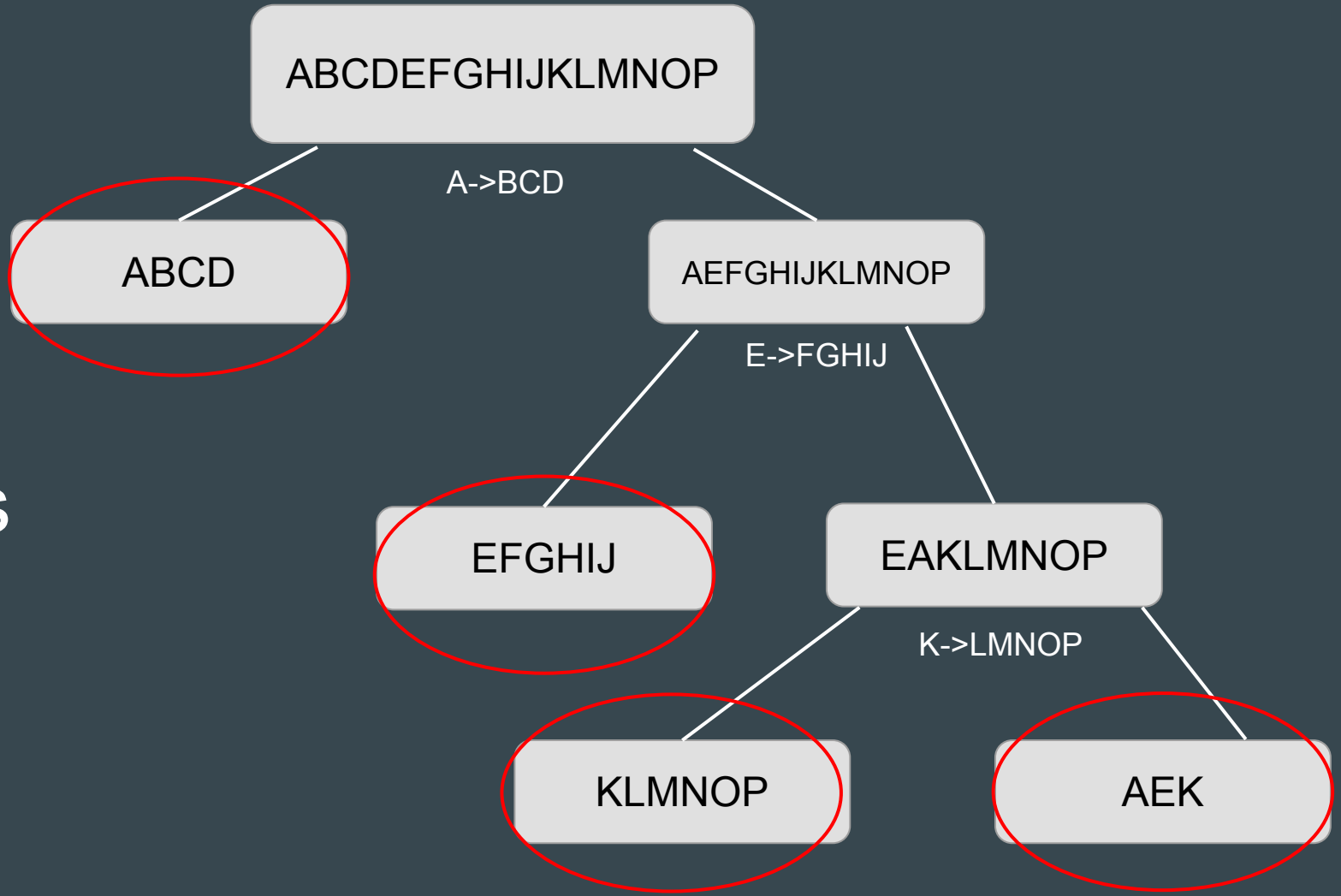
FD = { A->BCD,
E->FGHIJ,
K->LMNOP }

BCNF


ABCDEFGHIJKLMNOP
A->BCD
ABCD
AEFGHIJKLMNOP
E->FGHIJ
EFGHIJ
EAKLMNOP
K->LMNOP
KLMNOP
AEK

BCNF

Forms four tables



3NF

R= ABCDEFGHIJKLMNOP

FD = { A->BCD,
E->FGHIJ,
K->LMNOP }

Remove RHS Singletons

A->B
A->C
A->D
E->F
E->G
E->H
E->I
E->J
K->L
K->M
K->N
K->O
K->P

3NF

Remove Extraneous Attributes &
Remove Redundant Functional Dependencies

A->B
A->C
A->D
E->F
E->G
E->H
E->I
E->J
K->L
K->M
K->N
K->O
K->P

Merge FD's with same LHS

A->BCD
E->FGHIJ
K->LMNOP

3NF

Make Tables

R1 = ABCD
R2 = EFGHIJ
R3 = KLMNOP

No Subset Tables, so Check for Losslessness

Must make R* to match all attributes from every table

R1 = ABCD
R2 = EFGHIJ
R3 = KLMNOP
R* = AEK

AEK is the global key

Comparisons

- Our UML gives us 5 tables, BCNF and 3NF give us 4 tables.
- BCNF and 3NF gave us one table for what the store sells and what the customer is adding to their shopping cart.
- We chose to follow the UML design when creating our tables, which separates what the store sells and what the customer adds to their shopping cart into two different tables.
- This was simply for clarity purposes and helped us organize our data more effectively.

Schema

```
CREATE DOMAIN Rating integer
    Check (Value IN (1,2,3,4,5));

CREATE TABLE Store(
  storeid integer NOT NULL,
  sname character varying(255) NOT NULL,
  scity character varying(255) NOT NULL,
  shours character varying(255) NOT NULL,
  srating Rating NOT NULL,
  hiring boolean NOT NULL,
  CONSTRAINT store_pkey PRIMARY KEY (storeid)
);
```

```
CREATE TABLE Customer(
  custid integer NOT NULL,
  cfname character varying(255) NOT NULL,
  clname character varying(255) NOT NULL,
  ccity character varying(255) NOT NULL,
  username character varying(255) NOT NULL,
  password character varying(255) NOT NULL,
  CONSTRAINT customer_pkey PRIMARY KEY (custid)
);
```

```
CREATE TABLE ProductWarehouse(
  sku character varying(255) NOT NULL,
  pname character varying(255) NOT NULL,
  price real NOT NULL,
  category character varying(255) NOT NULL,
  CONSTRAINT products_pkey PRIMARY KEY (sku)
);
```

```
CREATE TABLE AddToCart(
  custid integer NOT NULL,
  sku varchar(255) NOT NULL,
  CONSTRAINT addtocart_pkey PRIMARY KEY (sku, custid),
  CONSTRAINT addtocart_custid_fkey FOREIGN KEY (custid)
      REFERENCES Customer (custid) MATCH SIMPLE
      ON UPDATE Cascade ON DELETE NO ACTION,
  CONSTRAINT addtocart_sku_fkey FOREIGN KEY (sku)
      REFERENCES ProductWarehouse (sku) MATCH SIMPLE
      ON UPDATE Cascade ON DELETE Cascade
);
```

```
CREATE TABLE StoreSells(
  sku character varying(255) NOT NULL,
  storeid integer NOT NULL,
  CONSTRAINT storesells_pkey PRIMARY KEY (sku, storeid),
  CONSTRAINT storesells_sku_fkey FOREIGN KEY (sku)
      REFERENCES ProductWarehouse (sku) MATCH SIMPLE
      ON UPDATE Cascade ON DELETE Cascade,
  CONSTRAINT storesells_storeid_fkey FOREIGN KEY (storeid)
      REFERENCES Store (storeid) MATCH SIMPLE
      ON UPDATE Cascade ON DELETE NO ACTION
);
```

Inserts

```
INSERT INTO Customer VALUES
(100,'Marcello','Anderson','Athens','mand4','qrs454'),
(106,'Keegan','Francis','Watkinsville','kef9','turtle3'),
(114,'Tristan','Earl','Atlanta','atlman3','braves989'),
(122,'Seth','Mac','Winder','setmac','ilovegroceries'),
(129,'Debbie','Monday','Macon','manicmonday','abc838');
```

```
INSERT INTO ProductWarehouse VALUES
(100,'Bag of Apples',18.59,'Produce'),
(118,'Chicken Breast',10.86,'Meat'),
(127,'Skim Milk',2.41,'Dairy'),
(136,'Sourdough Loaf',16.16,'Bakery'),
(157,'Merlot',15.72,'Alcohol'),
(199,'Eggo Waffles',13.87,'Frozen');
```

```
INSERT INTO StoreSells VALUES
(5012, 100),
(5003, 113),
(5011, 120),
(5012, 146),
(5029, 127),
(5026, 142);
```

```
INSERT INTO Store VALUES
(5000,'Kroger','Athens','24 hours',5,0),
(5006,'Bi-Lo','Watkinsville','24 hours',2,1),
(5012,'Trader Joes','Atlanta','10-10',4,0),
(5019,'Costco','Winder','7-10',5,0),
(5027,'Kroger','Macon','24 hours',5,1);
```

Queries

List all stores in a customer's city

```
Select Store.sname
From Store, Customer
Where Store.scity = Customer.ccity
AND Customer.ccity = 'Athens';
```

List stores that sell grapes

```
Select Store.name
From Store, StoreSells, Products
Where Store.storeid = StoreSells.storeid
AND StoreSells.sku = Products.sku
AND Products.name = 'Grapes';
```

List stores that sell all items in Customer 105's "cart"

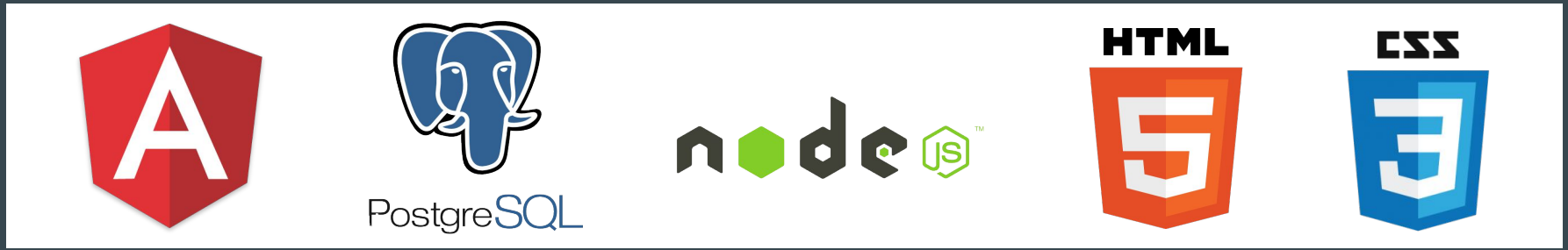
```
Select s.name
From Store s
Where not exists (
    Select ac.sku
    From AddToCart ac
    Where custid = 105 except (
        Select ss.sku
        From StoreSells ss
        Where s.storeid = ss.storeid)
);
```

List purchase info of customer with ID 113

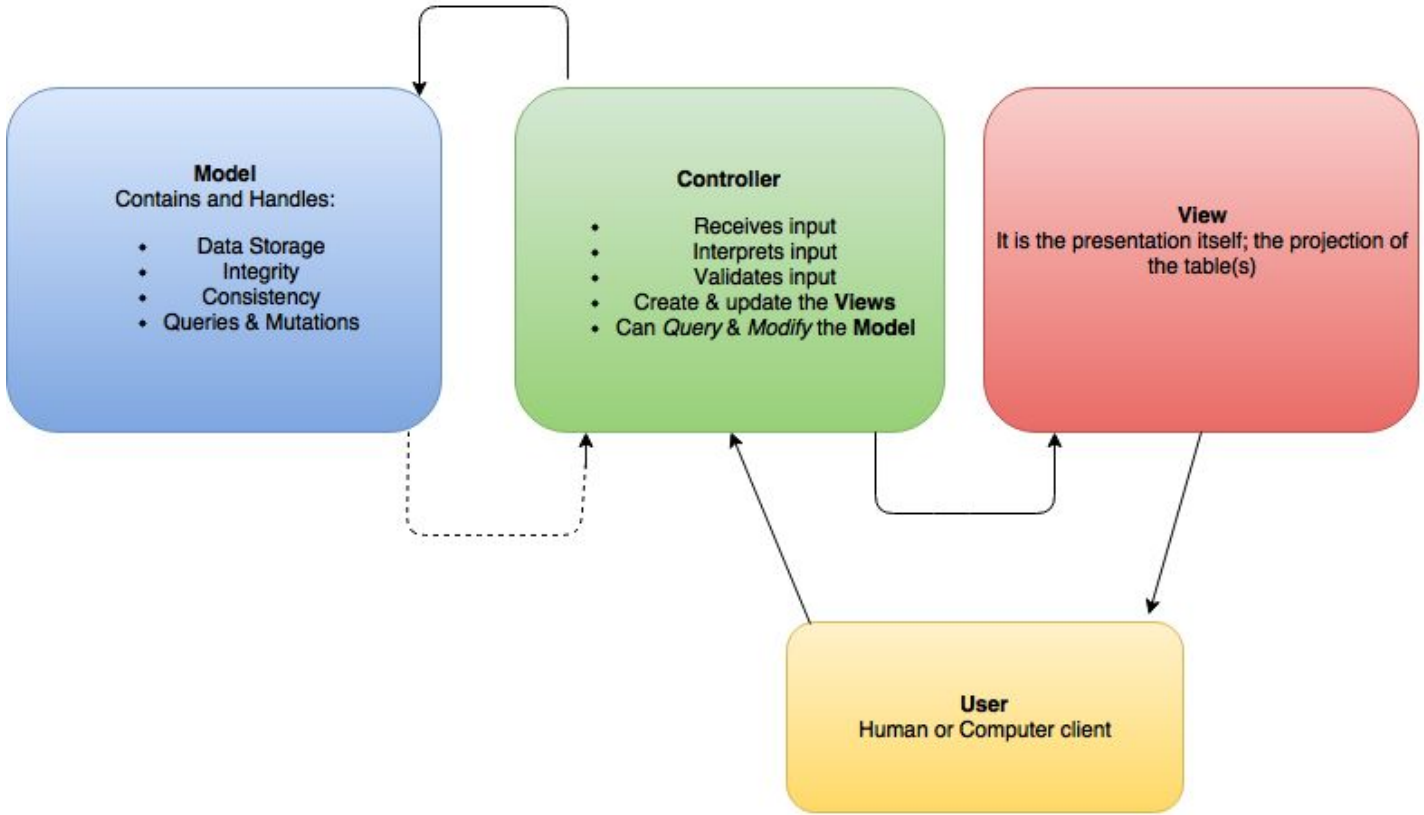
```
Select *
From AddToCart,Customer
Where AddToCart.custid = Customer.custid
AND Customer.custid = 113;
```

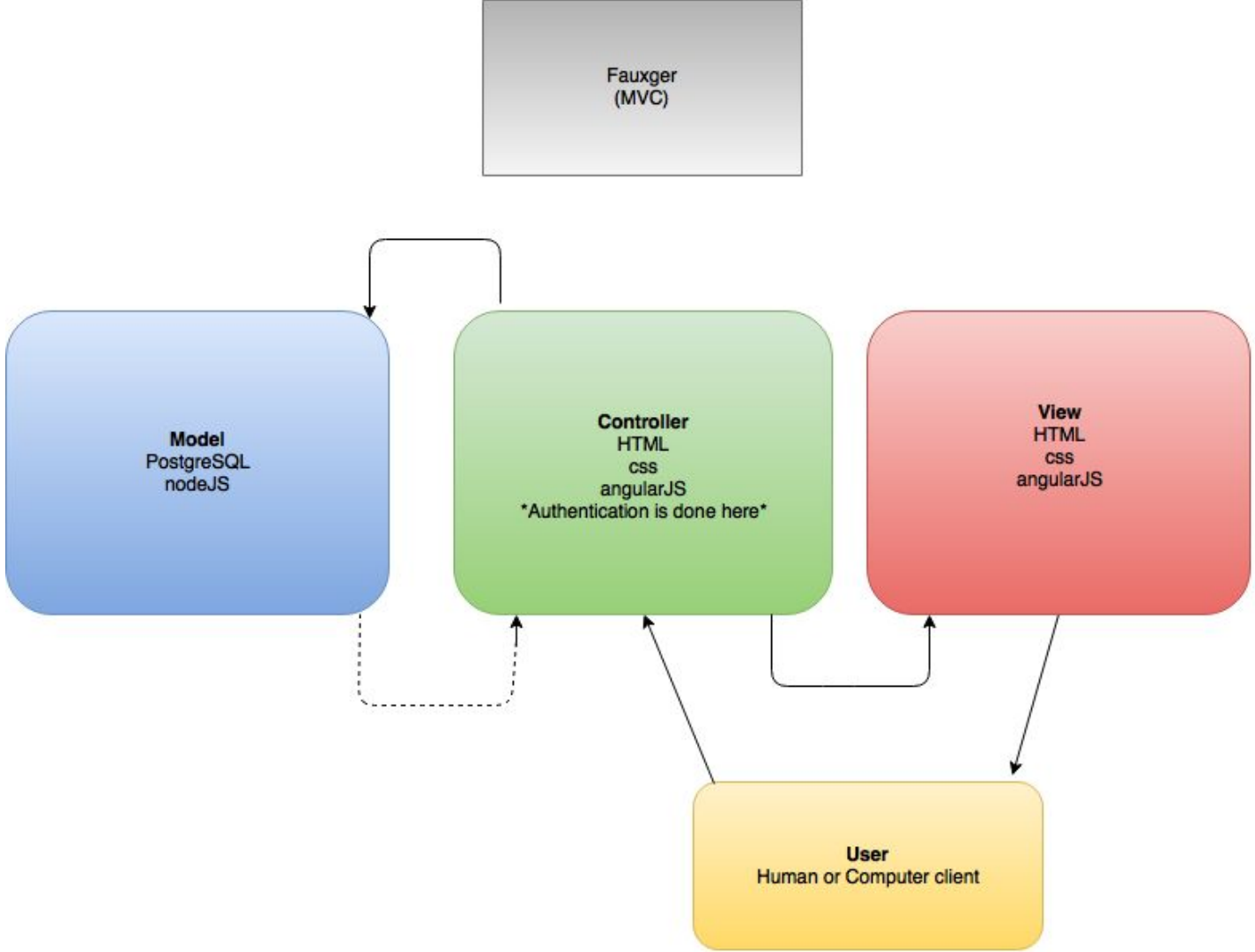
Architecture + Components

- Back-End: PostgreSQL, Node.js
- Front-End: HTML, CSS, AngularJS
- Security: Encrypted Passwords



Basics of Model View Controller (MVC)




Fauxger
(MVC)
Model
PostgreSQL
nodeJS
Controller
HTML
css
angularJS
Authentication is done here
View
HTML
css
angularJS
User
Human or Computer client