S2L: Eller EDMer's Chewy

We're far from purrrrfect, but welcome to our mini-world!

Today's Agenda
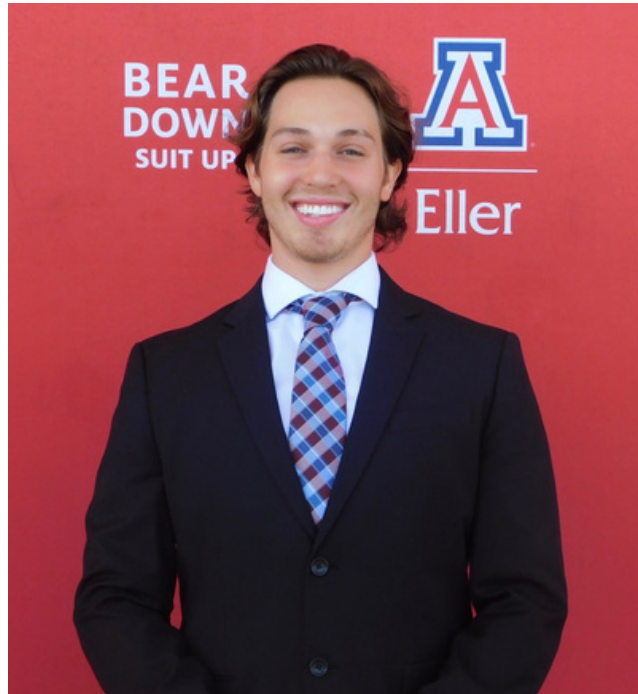
chewy

- Our Team
- Project Overview & Business Needs
- ER Diagrams
- Normalized Tables
- Checks & Constraints
- Queries
- Demo

Speaker: Tanishka

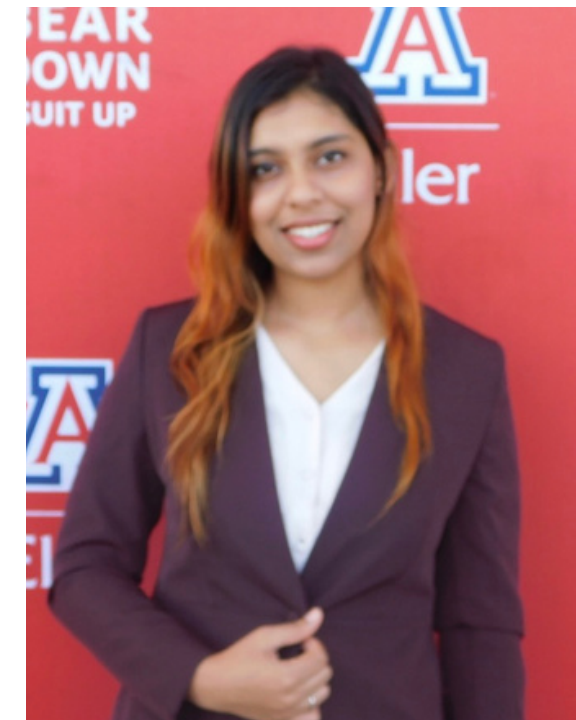# Eller's EDMers Team

chewy

Maximillian Goswitz

Unnati Palande

Bhargavi Somanahalli

3

Jaimin Fojdar

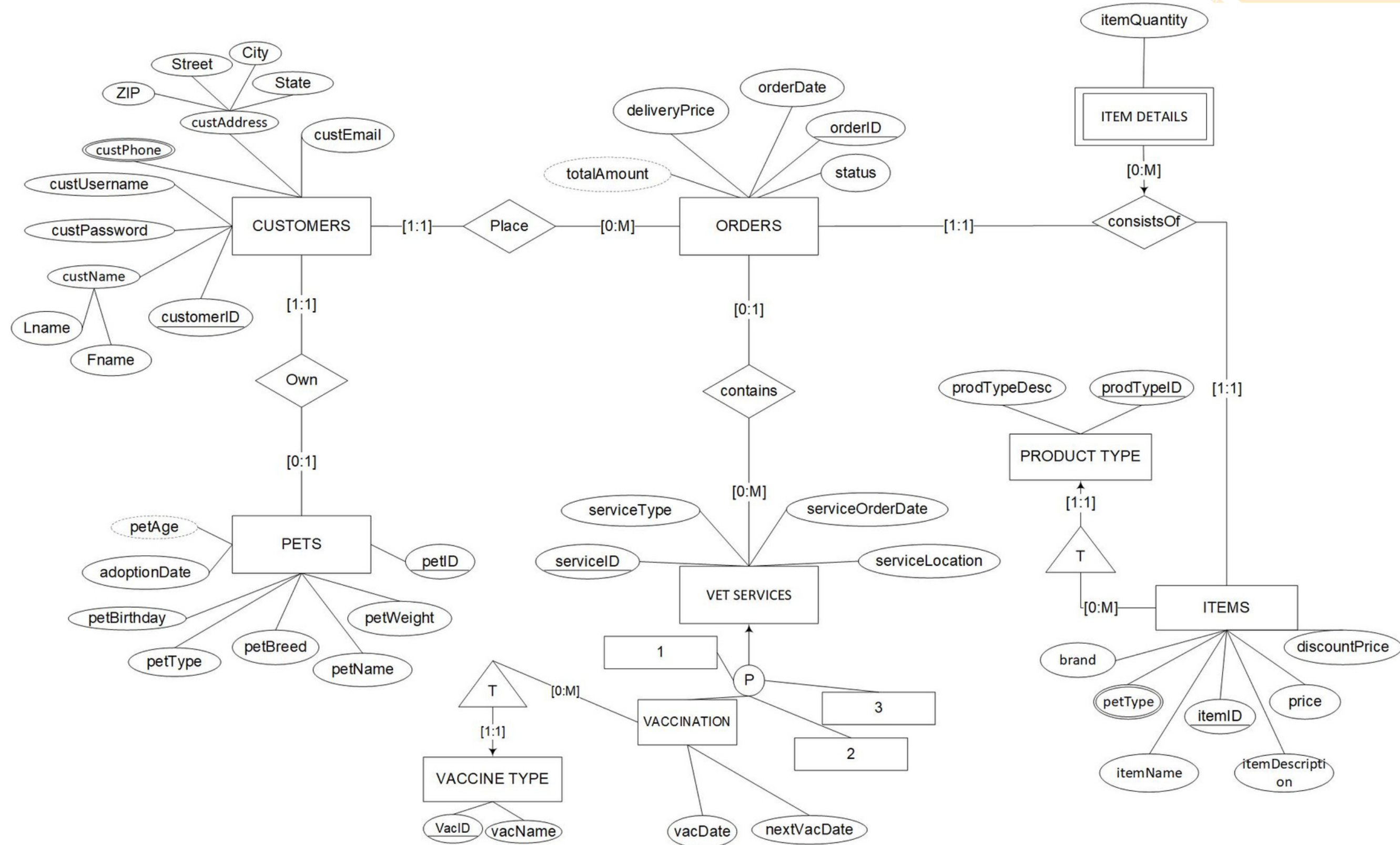Tanishka Shorey

Medha Singh

Overview

CHEWY's Mission Statement - To be the most trusted and convenient destination for pet parents (and partners), everywhere!

Creating an **Admin Dashboard** for CHEWY with the following functionalities:

- Onboarding new employees
- Analysing product sales
- Applying product line discounts
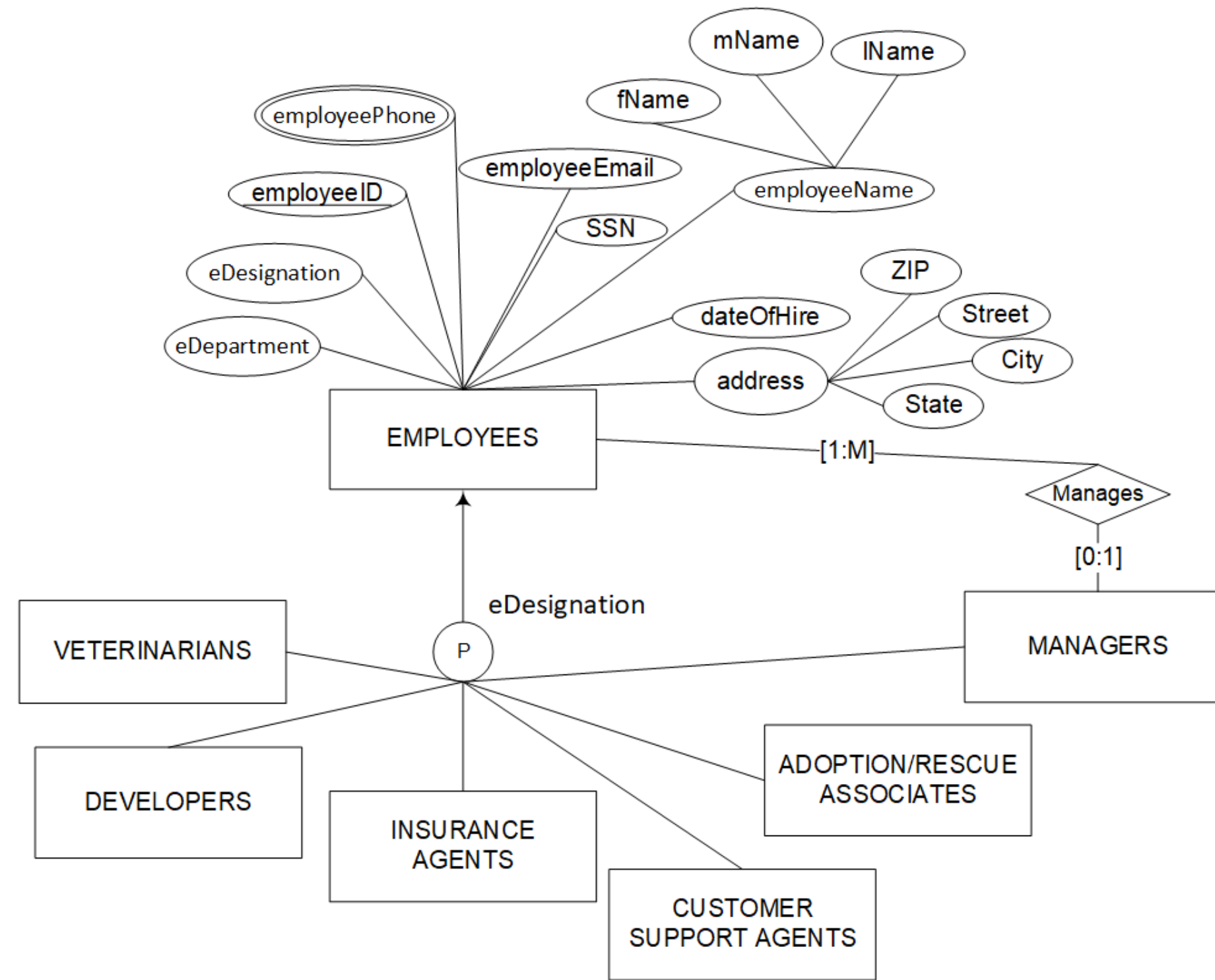- Tracking pet vaccinations

Speaker: Tanishka

chewy

Speaker: Max

Speaker: Max

Normalization

VET_SERVICES (<u>serviceID</u>, serviceType, serviceLocation, serviceOrderDate, orderID)
FOREIGN KEY (orderID) References **ORDERS** (orderID)

      VACCINATIONS (<u>serviceID</u>, vacID, vacName, vacDate, vacType, nextDue)
      FOREIGN KEY (serviceID) References **VET_SERVICES**(serviceID)

- **FD1**: serviceID --> vacID, vacName, vacDate, vacType, nextDue
- **FD2**: vacID --> vacName

      VACCINATIONS (<u>serviceID</u>, vacID, vacDate, vacType, nextDue)
      FOREIGN KEY (serviceID) References **VET_SERVICES**(serviceID)
      FOREIGN KEY (vacID) References **VACCINATION_NAMES**(vacID)

      VACCINATION_TYPES (<u>vacID</u>, vacName)

7

Checks & Constraints

chewy

EMPLOYEE PASSWORD

```
"REGEXP_COUNT(PASSWORD, '[0-9]') >= 1    -- 1 digit
     AND REGEXP_COUNT(PASSWORD, '[A-Z]') >= 1 -- 1 or more uppercase characters
     AND REGEXP_COUNT(PASSWORD, '[a-z]') >= 1 -- 1 or more lowercase characters
     AND LENGTH(PASSWORD) >= 8"
```

EMPLOYEE DEPARTMENT & DESIGNATION

```
constraint desig CHECK (designation IN ('ASSOCIATE', 'LEAD', 'MANAGER',
'CEO', 'CFO', 'CTO', 'CIO')),
constraint depart CHECK (department IN ('ADMIN', 'IT', 'SALES', 'HR',
'MARKETING'))
```
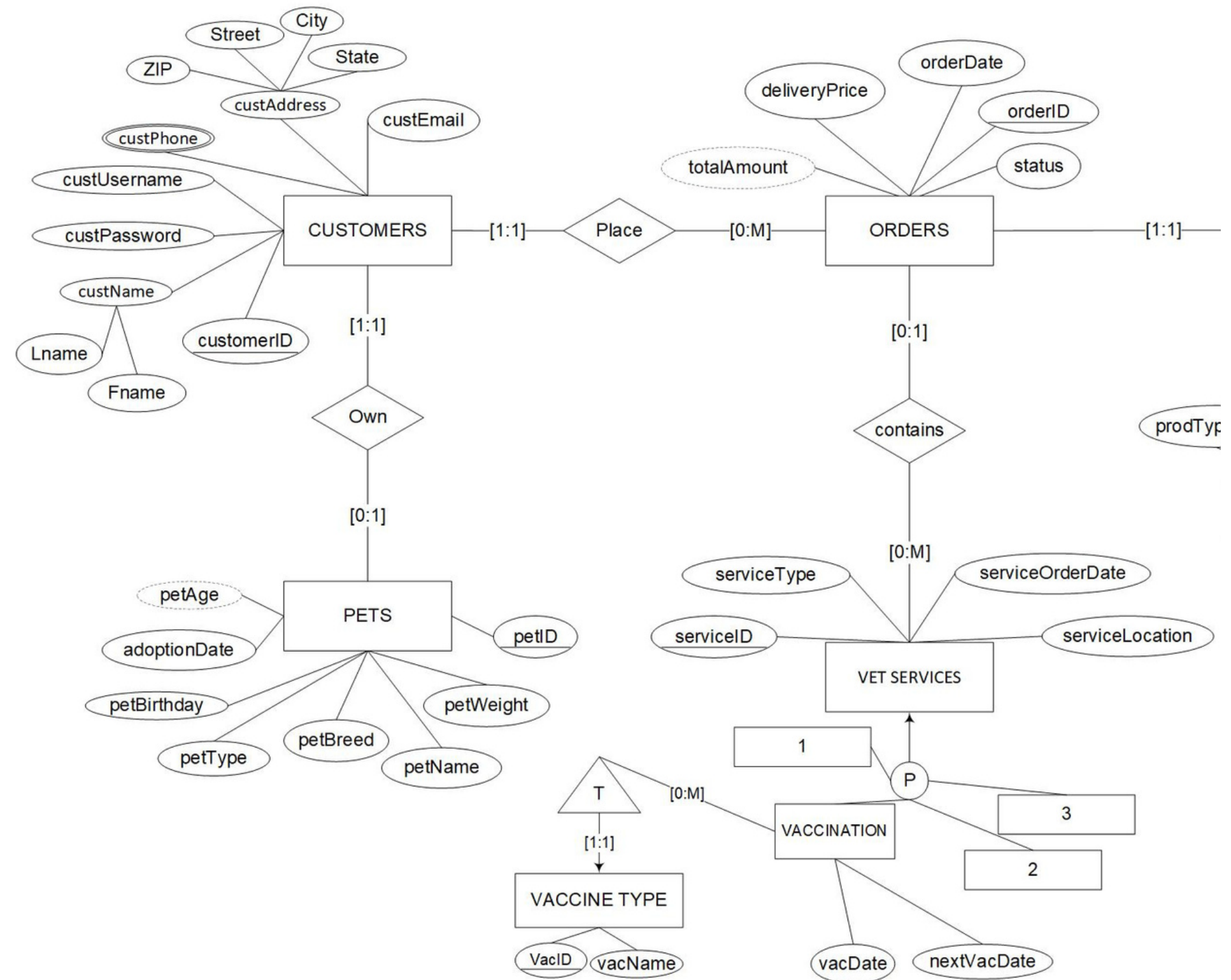
EMPLOYEE SSN
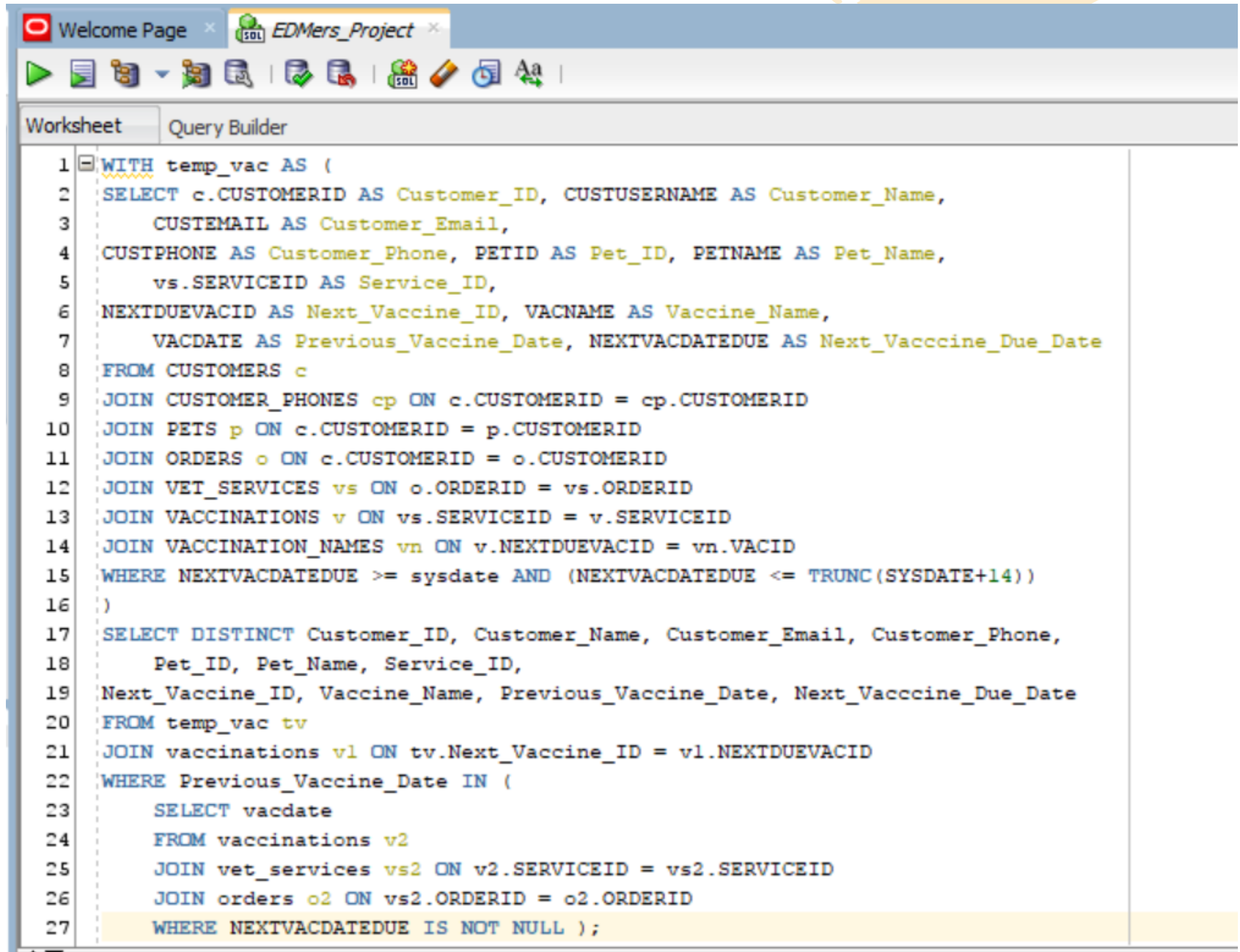
```
SSN char(9) constraint ssn_un UNIQUE
```

<u>Speaker</u>: Bhargavi

chewy

Vaccination Tracker

Speaker: Jai

SQL Query

chewy

Vaccination Tracker


```sql
WITH temp_vac AS (
SELECT c.CUSTOMERID AS Customer_ID, CUSTUSERNAME AS Customer_Name,
    CUSTEMAIL AS Customer_Email,
CUSTPHONE AS Customer_Phone, PETID AS Pet_ID, PETNAME AS Pet_Name,
    vs.SERVICEID AS Service_ID,
NEXTDUEVACID AS Next_Vaccine_ID, VACNAME AS Vaccine_Name,
    VACDATE AS Previous_Vaccine_Date, NEXTVACDATEDUE AS Next_Vaccine_Due_Date
FROM CUSTOMERS c
JOIN CUSTOMER_PHONES cp ON c.CUSTOMERID = cp.CUSTOMERID
JOIN PETS p ON c.CUSTOMERID = p.CUSTOMERID
JOIN ORDERS o ON c.CUSTOMERID = o.CUSTOMERID
JOIN VET_SERVICES vs ON o.ORDERID = vs.ORDERID
JOIN VACCINATIONS v ON vs.SERVICEID = v.SERVICEID
JOIN VACCINATION_NAMES vn ON v.NEXTDUEVACID = vn.VACID
WHERE NEXTVACDATEDUE >= sysdate AND (NEXTVACDATEDUE <= TRUNC(SYSDATE+14))
)
SELECT DISTINCT Customer_ID, Customer_Name, Customer_Email, Customer_Phone,
    Pet_ID, Pet_Name, Service_ID,
Next_Vaccine_ID, Vaccine_Name, Previous_Vaccine_Date, Next_Vacccine_Due_Date
FROM temp_vac tv
JOIN vaccinations v1 ON tv.Next_Vaccine_ID = v1.NEXTDUEVACID
WHERE Previous_Vaccine_Date IN (
    SELECT vacdate
    FROM vaccinations v2
    JOIN vet_services vs2 ON v2.SERVICEID = vs2.SERVICEID
    JOIN orders o2 ON vs2.ORDERID = o2.ORDERID
    WHERE NEXTVACDATEDUE IS NOT NULL );
```

10   Speaker: Jai

Vaccine Tracker Output

chewy

	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_EMAIL	CUSTOMER_PHONE	PET_ID	PET_NAME	SERVICE_ID	NEXT_VACCINE_ID	VACCINE_NAME	PREVIOUS_VACCINE_DATE	NEXT_VACCCINE_DUE_DATE
1	C10020	asugirl	asugirl@gmail.com	3751112851	P10016	Shadow	S111	168	Rabies-Dogs-Booster	05-DEC-2021	05-DEC-2022
2	C10016	asuguy	asuguy@gmail.com	5327289543	P10012	Cookie	S112	144	Rabies-Cats-Booster	07-DEC-2021	07-DEC-2022

Speaker: Jai

PL/SQL Procedure

Apply
Discount

```
CREATE OR REPLACE PROCEDURE discountprice (
    discount       NUMBER,
    in_brand       items.brand%TYPE,
    in_prodtype    product_types.prodtype%TYPE
) AS
    CURSOR cl IS
    SELECT itemid, price, brand, prodtype
    FROM items i JOIN product_types pt ON i.prodtypeid = pt.prodtypeid
    WHERE  brand = in_brand AND prodtype = in_prodtype;
    counttype           NUMBER;
    temp_itemid         items.itemid%TYPE;
    temp_brand          items.brand%TYPE;
    temp_prodtype       product_types.prodtype%TYPE;
    v_discountedprice   items.discountedprice%TYPE;
    temp_price          items.price%TYPE;
BEGIN
    SELECT COUNT(pt.prodtypeid) INTO counttype
    FROM items i JOIN product_types pt ON i.prodtypeid = pt.prodtypeid
    WHERE brand = in_brand AND prodtype = in_prodtype;
    IF counttype > 0 THEN
        FOR rec IN cl LOOP
            temp_itemid := rec.itemid;
            temp_brand := rec.brand;
            temp_prodtype := rec.prodtype;
            temp_price := rec.price;
            v_discountedprice := ( rec.price - ( rec.price * discount / 100 ) );
            UPDATE items
            SET
                discountedprice = v_discountedprice
            WHERE
                rec.itemid = itemid;
        END LOOP;
    ELSE
        raise_application_error(-20010, 'No Data Found');
    END IF;
END;
```

12   <u>Speaker</u>: Unnati

Apply Discount Output

chewy

BEFORE

	ITEMID	ITEMDESC	PRICE	DISCOUNTEDPRICE	BRAND	PRODTYPEID
1	I123	RopeToy	12	0	Tuffy	PT104

AFTER

	ITEMID	ITEMDESC	PRICE	DISCOUNTEDPRICE	BRAND	PRODTYPEID
1	I123	RopeToy	12	9.6	Tuffy	PT104

13   <u>Speaker</u>: Unnati

LIVE DEMO TIME!

Speaker: Medha

THANK YOU! QUESTIONS?