Question 5

Tables used in queries:

EMPLOYEE

```
CREATE TABLE EMPLOYEE
(
   EID INT PRIMARY KEY NOT NULL,
   ENAME VARCHAR(20) NOT NULL,
   EMPLOYMENTDATE DATE NOT NULL
)
```

CLOTHINGMODEL

```
CREATE TABLE CLOTHINGMODEL
(
   MODELNAME VARCHAR(30) PRIMARY KEY NOT NULL,
   SEX VARCHAR(1) NOT NULL,
   AGERANGE VARCHAR(10),
   MODELDISCOUNT INT DEFAULT 0,
   PRICE FLOAT(53) NOT NULL,
   CATID INT NOT NULL,
   CONSTRAINT CLOTHINGMODEL_CLOTHINGCATEGORY_CATID_FK FOREIGN KEY (CATID) REFERENCES
CLOTHINGCATEGORY (CATID)
);
COMMENT ON COLUMN CLOTHINGMODEL.MODELDISCOUNT IS 'Percentage Values'
```

CARTDETAILS

```
CREATE TABLE CARTDETAILS
(
   CARTID INT NOT NULL,
   COLOR VARCHAR(15) NOT NULL,
   SIZE VARCHAR(10) NOT NULL,
   MODELNAME VARCHAR(30) NOT NULL,
   QUANTITY INT DEFAULT 1 NOT NULL,
   CONSTRAINT CARTDETAILS_CARTID_COLOR_SIZE_MODELNAME_PK PRIMARY KEY (CARTID, COLOR, SIZE,
MODELNAME),
   CONSTRAINT CARTDETAILS_SHOPPINGCART_CARTID_FK FOREIGN KEY (CARTID) REFERENCES SHOPPINGCART
(CARTID),
   CONSTRAINT CARTDETAILS_CLOTHINGUNIT_COLOR_SIZE_MODELNAME_FK FOREIGN KEY (SIZE) REFERENCES
CLOTHINGUNIT (SIZE)
)
```

CLOTHINGCATEGORY

```
CREATE TABLE CLOTHINGCATEGORY
(
   CATID INT PRIMARY KEY NOT NULL,
   CATNAME VARCHAR(50) NOT NULL,
```

```
    CATTYPE VARCHAR(20) NOT NULL,
    CATEGORYDISCOUNT INT DEFAULT 0 NOT NULL
);
COMMENT ON COLUMN CLOTHINGCATEGORY.CATEGORYDISCOUNT IS 'Displayed as percentage'
```

ORDER

```
CREATE TABLE ORDER
(
    ORDERID INT PRIMARY KEY NOT NULL,
    ORDERTYPE VARCHAR(10) NOT NULL,
    PAYMENTMETHOD VARCHAR(20) NOT NULL,
    ORDERDATE DATE DEFAULT CURRENT DATE NOT NULL,
    FINALAMOUNT FLOAT(53) NOT NULL,
    CUSTOMEREMAIL VARCHAR(100) NOT NULL,
    HANDLEDATE DATE,
    HANDLER INT,
    SHIPPINGID INT,
    BILLINGID INT,
    CARTID INT,
    TRACKINGNUMBER VARCHAR(13),
    CONSTRAINT ORDER_CUSTOMER_EMAIL_FK FOREIGN KEY (CUSTOMEREMAIL) REFERENCES CUSTOMER
(EMAIL),
    CONSTRAINT ORDER_EMPLOYEE_EID_FK FOREIGN KEY (HANDLER) REFERENCES EMPLOYEE (EID),
    CONSTRAINT ORDER_SHIPPINGADDRESS_SA_ID_FK FOREIGN KEY (SHIPPINGID) REFERENCES
SHIPPINGADDRESS (SA_ID),
    CONSTRAINT ORDER_BILLINGADDRESS_BA_ID_FK FOREIGN KEY (BILLINGID) REFERENCES BILLINGADDRESS
(BA_ID),
    CONSTRAINT ORDER_SHOPPINGCART_CARTID_FK FOREIGN KEY (CARTID) REFERENCES SHOPPINGCART
(CARTID),
    CONSTRAINT ORDER_SHIPPINGINFO_TRACKINGNUMBER_FK FOREIGN KEY (TRACKINGNUMBER) REFERENCES
SHIPPINGINFO (TRACKINGNUMBER)
);
COMMENT ON COLUMN ORDER.ORDERTYPE IS 'PURCHASE or REFUND';
COMMENT ON COLUMN ORDER.PAYMENTMETHOD IS 'VISA/MASTERCARD/AMERICAN
EXPRESS/PAYPAL/INTERAC';
COMMENT ON COLUMN ORDER.SHIPPINGID IS 'Shipping Address ID';
COMMENT ON COLUMN ORDER.BILLINGID IS 'Billing Address ID';
COMMENT ON COLUMN ORDER.CARTID IS 'Shopping Cart ID'
```

UNITSTOCKING

```
CREATE TABLE UNITSTOCKING
(
    MODELNAME VARCHAR(30) NOT NULL,
    QUANTITYAVAILABLE INT NOT NULL,
    LOCATION VARCHAR(20) NOT NULL,
    COLOR VARCHAR(15) NOT NULL,
    SIZE VARCHAR(10) NOT NULL,
    CONSTRAINT UNITSTOCKING_COLOR_SIZE_PK PRIMARY KEY (COLOR, MODELNAME, SIZE, LOCATION),
    CONSTRAINT UNITSTOCKING_WAREHOUSE_LOCATION_FK FOREIGN KEY (LOCATION) REFERENCES
WAREHOUSE (LOCATION)
```

```
);
COMMENT ON COLUMN UNITSTOCKING.QUANTITYAVAILABLE IS 'Current Stock'
```

Queries

1) Description: This query selects all the employees that have started working from the date 2016-01-01 or later

```
SELECT *FROM EMPLOYEE WHERE EMPLOYMENTDATE >= DATE '2016-01-01';
```

	EID	ENAME	EMPLOYMENTDATE	SALARY
1	2435	Jeanne-Marie Gagnon	2016-12-12	74539
2	3654	Macaulay Mcguire	2016-08-29	71160
3	4525	Molly Conley	2016-02-18	188977
4	4881	Grace Dixon	2016-07-28	121275
5	1903	Ariel Bryan	2016-04-10	184335
6	1374	Cecilia Fleming	2016-08-13	142037
7	5011	Dai Gonzalez	2017-02-05	123421
8	5007	Paula Fox	2017-01-04	49055
9	5008	Jena Grimes	2017-01-07	69523
10	5009	Upton Mclean	2017-01-29	163472
11	5010	Rinah Morgan	2017-02-10	162589

2) Description: This query selects all the items from the carts of id 5000 and 5007, and outputs what items are in each cart. The query is an inner join of order, clothing model, and cart details.

```
SELECT CLOTHINGMODEL.PRICE, MYORDERID, MYCARTID, MYORDERTYPE, MYORDERDATE, MYCOLOR,
MYMODELNAME, MYSIZE
  FROM CLOTHINGMODEL INNER JOIN (
    SELECT ORDER.ORDERID AS MYORDERID, ORDER.CARTID AS MYCARTID, ORDER.ORDERTYPE AS
MYORDERTYPE, ORDER.ORDERDATE AS MYORDERDATE, CARTDETAILS.COLOR AS MYCOLOR,
CARTDETAILS.MODELNAME AS MYMODELNAME, CARTDETAILS.SIZE AS MYSIZE
    FROM ORDER INNER JOIN CARTDETAILS
    ON ORDER.CARTID = CARTDETAILS.CARTID
    WHERE (ORDER.CARTID = 5000 OR ORDER.CARTID = 5007))
  ON CLOTHINGMODEL.MODELNAME = MYMODELNAME;
```

	PRICE	MYORDERID	MYCARTID	MYORDERTYPE	MYORDERDATE	MYCOLOR	MYMODELNAME	MYSIZE
1	99.99	1000	5000	PURCHASE	2016-03-28	BLUE	Adidas Gloro 16.1 FG	40
2	99.99	1000	5000	PURCHASE	2016-03-28	YELLOW	Adidas Gloro 16.1 FG	44
3	99.99	1000	5000	PURCHASE	2016-03-28	RED	Adidas Gloro 16.1 FG	42
4	45	1000	5000	PURCHASE	2016-03-28	BLUE	Nike Tiempo Rio III FG	40
5	45	1000	5000	PURCHASE	2016-03-28	GREEN	Nike Tiempo Rio III FG	46
6	429.99	1000	5000	PURCHASE	2016-03-28	BLACK	Canada Goose Women	M
7	429.99	1007	5007	PURCHASE	2017-01-07	GREEN	Canada Goose Women	M
8	429.99	1000	5000	PURCHASE	2016-03-28	RED	Canada Goose Women	M
9	28.59	1007	5007	PURCHASE	2017-01-07	BLUE	Summer Breeze 2017	M
10	28.59	1007	5007	PURCHASE	2017-01-07	YELLOW	Summer Breeze 2017	L
11	28.59	1007	5007	PURCHASE	2017-01-07	GREEN	Summer Breeze 2017	M
12	40.99	1007	5007	PURCHASE	2017-01-07	BLUE	Slim Fit	M
13	40.99	1007	5007	PURCHASE	2017-01-07	WHITE	Slim Fit	L
14	40.99	1007	5007	PURCHASE	2017-01-07	BLUE	Slim Fit	S

3) Description: This query selects all the category names from clothing category whose catid is not in our subquery. Our subquery selects all clothing categories with ID's between 1000 and 4000, so our final query will select all clothing categories NOT between 1000 and 4000.

```
SELECT CATNAME FROM CLOTHINGCATEGORY WHERE CATID NOT IN (SELECT CATID FROM
CLOTHINGMODEL
  WHERE CATID BETWEEN 1000 and 4000);
```

	CATNAME
1	Blazer
2	Dress
3	V neck
4	Running
5	Boots
6	Scarves
7	Gloves
8	Socks
9	Belts
10	Sunglasses
11	Jewelery
12	Suspenders
13	Beanies
14	Caps

4) Description: This query selects all the refunds that customers made and the average amount per refund

```
SELECT AVG(FINALAMOUNT) AS AVERAGEREFUND, COUNT(*) as COUNT
FROM ORDER WHERE ORDERTYPE = 'REFUND';
```

	AVERAGEREFUND	COUNT
1	334.0731999999999	50

5) Description: With this query, we can see the total stock we have in each warehouse location

```
SELECT SUM(QUANTITYAVAILABLE), LOCATION
FROM UNITSTOCKING
WHERE QUANTITYAVAILABLE > 0
GROUP BY LOCATION;
```

	1	LOCATION
1	5651	MONTREAL
2	4911	NEW YORK CITY
3	4492	PARIS
4	4324	TORONTO