**TERM PROJECT: BA840**

**Team:** All Stars

**Team Members:**

Yash Malhan Xi Gong Shadman Ashraf Shih-Hsun Lin

**Part 1**

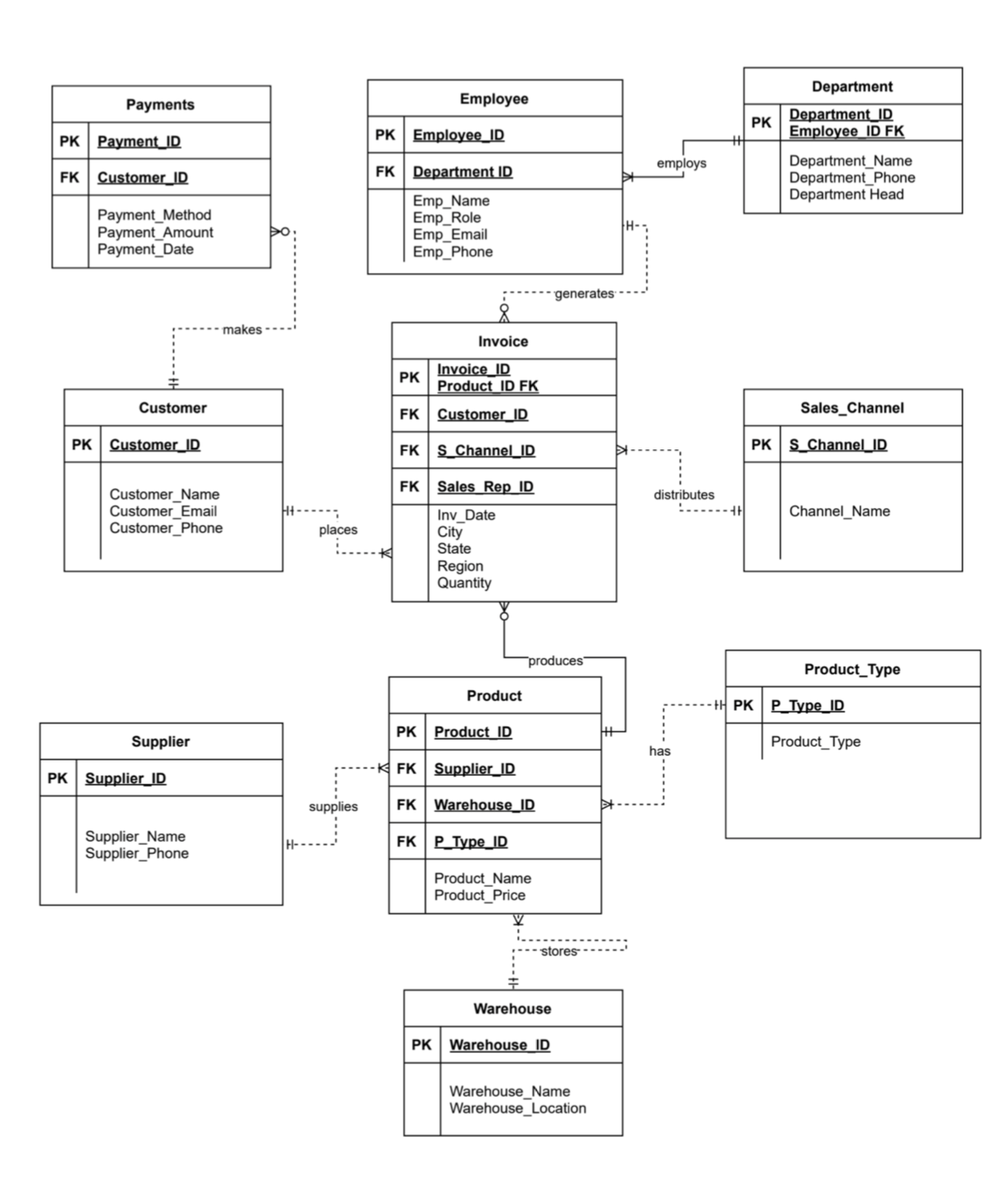
* The general system of this relational database is built upon the complete business cycle of a high-end office/digital supplies reseller, Allstar. The advantage of being a reseller is that Allstar is able to engage in both B2B and B2C transactions and build a diverse client base. Allstar utilizes this database to perform billing, stocking,distributing, HR and etc functions and therefore the database has several subsystems:

1. Customer subsystem stores the unique customer id which is automatically generated when our employee adds a customer to our system or when a customer signs up online. For privacy protection, we only record customer first name and essential contact information.
2. Employee subsystem stores employees’ personal, organizational, and contact information. Allstar has a special incentive program to foster a competitive working environment in which all employees, regardless of duties and level, are encouraged to make sales and enlarge our clients network and earn commissions accordingly. Historically we have seen employees on driver/warehouse workers making huge sales.
3. Department sub-system stores each department’s information and all of its employees' unique ids. Allstar does not have traditional department names, it only has numbers because each department has a mix of all functional employees and are fully rotational.
4. Supplier sub-system records the supplier’s unique ID(not replaceable) name, and contact information. Brand associate is in charge of maintaining supplier relationships and evaluating contract renewal. Abandoned suppliers are permanently dropped from the system.
5. Warehouse subsystem records warehouse basic and geographical information. Allstars’ warehouses have been stable and do not expect expansion until a significant business scale update.
6. Product type subsystem stores the product types Allstar intends to source and sell after thorough market research and discussions by upper management.
7. Product subsystem records the unique id of each product(not replaceable), general product information, where it comes from and where it is stored. Outdated products will be permanently dropped from the system.
8. Sales channel subsystem records the channels Allstar list or distributes its products.
9. Invoice subsystem, records unique invoice id, customer information, timestamp, relevant salesperson information every time a sale has been made. Invoice ID start with the last two digits of the current year, and the will be zeroed at the beginning of a new fiscal year. Delivery address is not directly recorded in this subsystem due to privacy protection, and segregation of duties reasons. Instead, delivery addresses can be hashed using the combination of invoice id, city, and state as keys in a security sandbox outside Allstar’s primary system.
10. Payment subsystem records the customer id, payment method, amount, timestamp every time a payment has been made. A unique payment ID will also be automatically generated. Payment method cash records not just bills but also cash equivalents such as money order, check, anc online echeck, essentially any payment that is not on account. Allstar’s customers are mostly on payment terms/plans so payments are usually not in full amount.

**Part 2**

* Customer-Payments: Customer table has an optional one to many relationship with the payments table. A customer may make zero multiple payments but each payment corresponds to only one customer.
* Customer-Invoice: Customer table has a one to many relationship with the invoice table. A customer may have multiple invoices under their name but each invoice corresponds to only one customer.
* Invoice-Employee: Invoice table has an optional one to many relationship with the employee table. An employee/sales representative may generate zero to multiple invoices but each invoice corresponds to only one employee/sales representative.
* Department-Employee: Department table has a one to many relationship to the employee table. Each department may have multiple employees but one employee can only be employed in one department.
* Sales\_Channel-Invoice: Sales\_channel table has a one to many relationship with the invoice table. One sales channel may distribute multiple invoice/product combinations, but each invoice/product combination can only be distributed in one particular sales channel.
* Invoice-Product: Invoice table has an optional one to many relationship with the product table. A product can be found on zero to multiple invoice/product combinations, but each invoice/product combination can only contain one product.
* Supplier-Product: Supplier table has one to many relationship with the product table. Each supplier can supply multiple products but we source each product from only one supplier.
* Product\_type-Product: Product\_type table has a one to many relationship with the product table. Each product type can have multiple products, but each product can only be in one particular product category.
* Warehouse-Product: Warehouse table has a one to many relationship with the product table. Each warehouse can store multiple products, but each product can only be stored in one warehouse at a time.

**Part 3**



**Part 4**

**Create & Alter Statements:**

CREATE TABLE TPCUSTOMER (

CUSTOMER\_ID INT PRIMARY KEY,

CUSTOMER\_NAME VARCHAR(255),

CUSTOMER\_EMAIL VARCHAR(255),

CUSTOMER\_PHONE VARCHAR(10));

ALTER TABLE TPCUSTOMER MODIFY COLUMN CUSTOMER\_PHONE VARCHAR(15);

CREATE TABLE TPWAREHOUSE(

WAREHOUSE\_ID INT PRIMARY KEY,

WAREHOUSE\_NAME VARCHAR(38),

WAREHOUSE\_LOCATION VARCHAR(50)

);

CREATE TABLE TPSUPPLIER (

SUPPLIER\_ID INT PRIMARY KEY,

SUPPLIER\_NAME VARCHAR(255) NOT NULL,

SUPPLIER\_PHONE VARCHAR(10)

);

ALTER TABLE TPSUPPLIER MODIFY COLUMN SUPPLIER\_PHONE VARCHAR(15);

CREATE TABLE TPDEPARTMENT(

DEPARTMENT\_ID INT,

EMPLOYEE\_ID INT,

DEPARTMENT\_HEAD VARCHAR(20),

DEPARTMENT\_NAME VARCHAR(25),

DEPARTMENT\_PHONE VARCHAR(10),

PRIMARY KEY (DEPARTMENT\_ID, EMPLOYEE\_ID)

);

CREATE TABLE TPPAYMENTS (

PAYMENT\_ID INT PRIMARY KEY,

CUSTOMER\_ID INT,

PAYMENT\_METHOD VARCHAR(10),

PAYMENT\_AMOUNT NUMERIC(20,2),

PAYMENT\_DATE DATE,

FOREIGN KEY (CUSTOMER\_ID) REFERENCES TPCUSTOMER(CUSTOMER\_ID)

);

CREATE TABLE TPPRODUCT\_TYPE(

P\_TYPE\_ID INT PRIMARY KEY,

PRODUCT\_TYPE VARCHAR(38));

CREATE TABLE TPPRODUCT (

PRODUCT\_ID INT PRIMARY KEY,

SUPPLIER\_ID INT,

WAREHOUSE\_ID INT,

P\_TYPE\_ID INT,

PRODUCT\_NAME VARCHAR(50) NOT NULL,

PRODUCT\_PRICE INT,

FOREIGN KEY (SUPPLIER\_ID) REFERENCES TPSUPPLIER(SUPPLIER\_ID),

FOREIGN KEY (WAREHOUSE\_ID) REFERENCES TPWAREHOUSE(WAREHOUSE\_ID),

FOREIGN KEY (P\_TYPE\_ID) REFERENCES TPPRODUCT\_TYPE (P\_TYPE\_ID));

DROP TABLE TPPRODUCT;

CREATE TABLE TPEMPLOYEE (

EMPLOYEE\_ID INT PRIMARY KEY,

DEPARTMENT\_ID INT,

EMPLOYEE\_NAME VARCHAR(20) NOT NULL,

EMPLOYEE\_ROLE VARCHAR(25) NOT NULL,

EMPLOYEE\_EMAIL VARCHAR(25),

EMPLOYEE\_PHONE VARCHAR(20),

FOREIGN KEY (DEPARTMENT\_ID) REFERENCES TPDEPARTMENT(DEPARTMENT\_ID)

);

CREATE TABLE TPSALES\_CHANNEL(

S\_CHANNEL\_ID INT,

CHANNEL\_NAME VARCHAR(38)

);

ALTER TABLE TPSALES\_CHANNEL ADD PRIMARY KEY(S\_CHANNEL\_ID);

CREATE TABLE TPINVOICE (

INVOICE\_ID INT,

PRODUCT\_ID INT,

CUSTOMER\_ID INT,

S\_CHANNEL\_ID INT,

SALES\_REP\_ID INT,

INV\_DATE DATE,

CITY VARCHAR(50),

STATE VARCHAR(50),

REGION VARCHAR (50),

QUANTITY INT

);

ALTER TABLE TPINVOICE ADD PRIMARY KEY (INVOICE\_ID, PRODUCT\_ID);

ALTER TABLE TPINVOICE ADD FOREIGN KEY(PRODUCT\_ID) REFERENCES TPPRODUCT(PRODUCT\_ID);

ALTER TABLE TPINVOICE ADD FOREIGN KEY(CUSTOMER\_ID) REFERENCES TPCUSTOMER(CUSTOMER\_ID);

ALTER TABLE TPINVOICE ADD FOREIGN KEY(SALES\_REP\_ID) REFERENCES TPEMPLOYEE (EMPLOYEE\_ID);

ALTER TABLE TPINVOICE ADD FOREIGN KEY(S\_CHANNEL\_ID) REFERENCES TPSALES\_CHANNEL(S\_CHANNEL\_ID);

**Insert Statements:**

INSERT INTO TPDEPARTMENT VALUES (1001, 83304, 'D1', 'Aaron Bergman', '453-9342');

INSERT INTO TPDEPARTMENT VALUES (1001, 83308,'D1', 'Aaron Bergman', '495-3082');

INSERT INTO TPDEPARTMENT VALUES (1001, 83312,'D1','Aaron Bergman', '632-8197');

INSERT INTO TPDEPARTMENT VALUES (1002, 83314,'D2','Darren Powers', '726-5086');

INSERT INTO TPDEPARTMENT VALUES (1003, 83318, 'D3','Michael Stewart','680-6124');

INSERT INTO TPDEPARTMENT VALUES (1001, 83321,'D1','Aaron Bergman', '734-7097');

INSERT INTO TPDEPARTMENT VALUES (1002, 83332,'D2','Darren Powers', '242-1979');

INSERT INTO TPDEPARTMENT VALUES (1001, 83341,'D1', 'Aaron Bergman', '508-3269');

INSERT INTO TPDEPARTMENT VALUES (1005, 83347,'D5','Brad Eason', '653-6753');

INSERT INTO TPDEPARTMENT VALUES (1004, 83349,'D4','Alan Schoenberger', '722-9727');

INSERT INTO TPDEPARTMENT VALUES (1002, 83359, 'D2','Darren Powers', '406-3691');

INSERT INTO TPDEPARTMENT VALUES (1002, 83366, 'D2','Darren Powers', '686-7916');

INSERT INTO TPDEPARTMENT VALUES (1002, 83371, 'D2','Darren Powers', '512-3569');

INSERT INTO TPDEPARTMENT VALUES (1001, 83372,'D1','Aaron Bergman', '326-1880');

INSERT INTO TPDEPARTMENT VALUES (1005, 83374,'D5', 'Brad Eason', '305-1308');

INSERT INTO TPDEPARTMENT VALUES (1006, 83378, 'D6', 'Justin Ritter', '255-9624');

INSERT INTO TPDEPARTMENT VALUES (1005, 83382, 'D5','Brad Eason', '511-5282');

INSERT INTO TPDEPARTMENT VALUES (1004, 83385, 'D4','Alan Schoenberger', '888-6311');

INSERT INTO TPDEPARTMENT VALUES (1003, 83398, 'D3','Michael Stewart', '342-9233');

INSERT INTO TPDEPARTMENT VALUES (1002, 83403,'D2', 'Darren Powers', '310-1479');

INSERT INTO TPEMPLOYEE VALUES (83304, 1001,'TAMARA MCDONALD', 'ASSOCIATE', 'T.MCDONA3@GMAIL.COM', '453-9342');

INSERT INTO TPEMPLOYEE VALUES (83308, 1001,'CONNIE LOVE', 'SENIOR ASSOCIATE', 'C.LOVE77@GMAIL.COM', '495-3082');

INSERT INTO TPEMPLOYEE VALUES (83312, 1001,'ROSALBA BAKER', 'ASSOCIATE', 'R.BAKER9@GMAIL.COM', '632-8197');

INSERT INTO TPEMPLOYEE VALUES (83314, 1002, 'CHAROLETTE DAVID', 'ASSOCIATE', 'C.DAVID84@GMAIL.COM', '726-5086');

INSERT INTO TPEMPLOYEE VALUES (83318, 1003,'DARCIE PECK', 'FREIGHT STOCKER', 'D.PECK4@GMAIL.COM', '680-6124');

INSERT INTO TPEMPLOYEE VALUES (83321, 1001,'ANGELINA FARMER', 'BRAND ASSOCIATE', 'A.FARMER86@GMAIL.COM', '734-7097');

INSERT INTO TPEMPLOYEE VALUES (83332, 1002,'WILLARD LONG', 'BRAND ASSOCIATE', 'W.LONG1@GMAIL.COM', '242-1979');

INSERT INTO TPEMPLOYEE VALUES (83341, 1001,'CHRISTINE CORTEZ', 'ASSOCIATE', 'C.CORTEZ85@GMAIL.COM', '508-3269');

INSERT INTO TPEMPLOYEE VALUES (83347, 1005,'QUINTIN WINN', 'SENIOR FREIGHTER', 'Q.WINN82@GMAIL.COM', '653-6753');

INSERT INTO TPEMPLOYEE VALUES (83349, 1004,'JENNIFFER SINGH', 'SENIOR FREIGHTER', 'J.SINGH87@GMAIL.COM', '722-9727');

INSERT INTO TPEMPLOYEE VALUES (83359, 1002,'MERLE WATTS', 'SENIOR ASSOCIATE', 'M.WATTS90@GMAIL.COM', '406-3691');

INSERT INTO TPEMPLOYEE VALUES (83366, 1002,'PHOEBE BLEDSOE', 'ASSOCIATE', 'P.BLEDSO99@GMAIL.COM', '686-7916');

INSERT INTO TPEMPLOYEE VALUES (83371, 1002,'ROXANE MATHEWS', 'FREIGHT STOCKER', 'R.MATHEW94@GMAIL.COM', '512-3569');

INSERT INTO TPEMPLOYEE VALUES (83372, 1001,'CLAUDINE DAHL', 'DRIVER', 'C.DAHL90@GMAIL.COM', '326-1880');

INSERT INTO TPEMPLOYEE VALUES (83374, 1005,'DARRON TILLEY', 'FREIGHT STOCKER', 'D.TILLEY1@GMAIL.COM', '305-1308');

INSERT INTO TPEMPLOYEE VALUES (83378, 1006,'FELICIA DUNHAM', 'ASSOCIATE', 'F.DUNHAM5@GMAIL.COM', '255-9624');

INSERT INTO TPEMPLOYEE VALUES (83382, 1005,'STELLA CONKLIN', 'ASSOCIATE', 'S.CONKLI80@GMAIL.COM', '511-5282');

INSERT INTO TPEMPLOYEE VALUES (83385, 1004,'BRODERICK COLBERT', 'FREIGHT STOCKER', 'B.COLBER91@GMAIL.COM', '888-6311');

INSERT INTO TPEMPLOYEE VALUES (83398, 1003,'ZACK GILES', 'FREIGHT STOCKER', 'Z.GILES9@GMAIL.COM', '342-9233');

INSERT INTO TPEMPLOYEE VALUES (83403, 1002,'FELICITA PONCE', 'FREIGHT STOCKER', 'F.PONCE79@GMAIL.COM', '310-1479');

INSERT INTO TPCUSTOMER VALUES(8240222,'Jack','Ja8779@gmail.com','105-970-8406');

INSERT INTO TPCUSTOMER VALUES(8240223,'Miles','Mi5237@gmail.com','906-798-6145');

INSERT INTO TPCUSTOMER VALUES(8240224,'Yash','Ya7261@gmail.com','995-998-6157');

INSERT INTO TPCUSTOMER VALUES(8240225,'Yuki','Yu8389@gmail.com','157-103-7423');

INSERT INTO TPCUSTOMER VALUES(8240250,'Amanda','Am7693@gmail.com','809-162-9721');

INSERT INTO TPCUSTOMER VALUES(8240226,'Shadman','Sh4353@gmail.com','596-621-5923');

INSERT INTO TPCUSTOMER VALUES(8240227,'Xi','Xi4233@gmail.com','163-440-3039');

INSERT INTO TPCUSTOMER VALUES(8240228,'Robert','Ro545@gmail.com','596-245-6381');

INSERT INTO TPCUSTOMER VALUES(8240229,'Jesse','Je3356@gmail.com','866-649-3731');

INSERT INTO TPCUSTOMER VALUES(8240230,'Ish','Is4006@gmail.com','981-147-6821');

INSERT INTO TPCUSTOMER VALUES(8240231,'Oyin','Oy28@gmail.com','301-157-8415');

INSERT INTO TPCUSTOMER VALUES(8240232,'Zoe','Zo4593@gmail.com','179-163-6246');

INSERT INTO TPCUSTOMER VALUES(8240233,'Rachel','Ra8900@gmail.com','606-843-5551');

INSERT INTO TPCUSTOMER VALUES(8240234,'BinBin','Bi1645@gmail.com','889-382-4957');

INSERT INTO TPCUSTOMER VALUES(8240235,'Shang','Sh8650@gmail.com','165-502-4819');

INSERT INTO TPCUSTOMER VALUES(8240236,'Hao','Ha8187@gmail.com','584-723-1608');

INSERT INTO TPCUSTOMER VALUES(8240237,'Gilly','Gi192@gmail.com','392-299-5001');

INSERT INTO TPCUSTOMER VALUES(8240238,'Feld','Fe271@gmail.com','982-831-2413');

INSERT INTO TPCUSTOMER VALUES(8240239,'Thomas','Th2205@gmail.com','373-114-2900');

INSERT INTO TPCUSTOMER VALUES(8240240,'Karen','Ka2552@gmail.com','962-153-2905');

INSERT INTO TPCUSTOMER VALUES(8240241,'Leslie','Le914@gmail.com','859-335-4688');

INSERT INTO TPCUSTOMER VALUES(8240242,'Travis','Tr8615@gmail.com','498-465-2302');

INSERT INTO TPCUSTOMER VALUES(8240243,'Oscar','Os716@gmail.com','249-516-9127');

INSERT INTO TPCUSTOMER VALUES(8240244,'Paul','Pa4470@gmail.com','305-566-1140');

INSERT INTO TPCUSTOMER VALUES(8240245,'Qi','Qi982@gmail.com','813-547-4845');

INSERT INTO TPCUSTOMER VALUES(8240246,'Ashley','As3000@gmail.com','599-843-9497');

INSERT INTO TPCUSTOMER VALUES(8240247,'Sung','Su2536@gmail.com','971-798-7007');

INSERT INTO TPCUSTOMER VALUES(8240248,'Peter','Pe6354@gmail.com','453-364-9847');

INSERT INTO TPCUSTOMER VALUES(8240249,'Steven','St8743@gmail.com','589-461-8699');

INSERT INTO TPSALES\_CHANNEL VALUES (1660,'BestBuy');

INSERT INTO TPSALES\_CHANNEL VALUES (1661,'IKEA');

INSERT INTO TPSALES\_CHANNEL VALUES (1662,'Costco');

INSERT INTO TPSALES\_CHANNEL VALUES (1666,'WalMART');

INSERT INTO TPSALES\_CHANNEL VALUES (1667,'Target');

INSERT INTO TPSALES\_CHANNEL VALUES (1668,'IKEA');

INSERT INTO TPSALES\_CHANNEL VALUES (1669,'STAPLE');

INSERT INTO TPSALES\_CHANNEL VALUES (1672,'Amazon');

INSERT INTO TPSALES\_CHANNEL VALUES (1676,'J.C.Penny');

INSERT INTO TPSALES\_CHANNEL VALUES (1682,'Lowe');

INSERT INTO TPSUPPLIER VALUES (2001, 'MARS TRADERS', '554-653-8624');

INSERT INTO TPSUPPLIER VALUES (2011, 'RIWITS', '534-673-8452');

INSERT INTO TPSUPPLIER VALUES (2012, 'SIMPHONIES', '534-852-1432');

INSERT INTO TPSUPPLIER VALUES (2019, 'MAKS', '555-745-7456');

INSERT INTO TPSUPPLIER VALUES (2098, 'JONAS SUPPLIES', '434-662-5367');

INSERT INTO TPPRODUCT\_TYPE VALUES (1201,'Office Supplies');

INSERT INTO TPPRODUCT\_TYPE VALUES (1202,'Computer&Accesories');

INSERT INTO TPPRODUCT\_TYPE VALUES (1203,'Furniture');

INSERT INTO TPPRODUCT\_TYPE VALUES (1204,'Multimedia');

INSERT INTO TPPRODUCT\_TYPE VALUES (1205,'General Appliance');

INSERT INTO TPPRODUCT\_TYPE VALUES (1206,'Storage');

INSERT INTO TPWAREHOUSE VALUES (101, 'ACTIVEWEAR WAREHOUSE', 'HOUSTON, TX');

INSERT INTO TPWAREHOUSE VALUES (102, 'TURTLE WAREHOUSE', 'COLUMBUS, OH');

INSERT INTO TPWAREHOUSE VALUES (103, 'EAST MANCHESTER WAREHOUSE', 'Los Angeles, CA');

INSERT INTO TPWAREHOUSE VALUES (104, 'ARSENAL WAREHOUSE', 'DETROIT MI');

INSERT INTO TPPAYMENTS VALUES (10001,8240222, 'MASTERCARD', 299.89, '2020-01-03');

INSERT INTO TPPAYMENTS VALUES (10002,8240224, 'DISCOVER', 669.29, '2020-01-24');

INSERT INTO TPPAYMENTS VALUES (10003,8240226, 'CASH', 299.89, '2020-01-27');

INSERT INTO TPPAYMENTS VALUES (10004,8240229, 'CASH', 599.19, '2020-02-08');

INSERT INTO TPPAYMENTS VALUES (10005,8240226, 'VISA', 229.89, '2020-02-10');

INSERT INTO TPPAYMENTS VALUES (10006,8240237, 'VISA', 399.99, '2020-02-23');

INSERT INTO TPPAYMENTS VALUES (10007,8240224, 'MASTERCARD', 100.00, '2020-03-03');

INSERT INTO TPPAYMENTS VALUES (10008,8240226, 'DISCOVER', 169.89, '2020-03-05');

INSERT INTO TPPAYMENTS VALUES (10009,8240229, 'CASH', 399.99, '2020-03-07');

INSERT INTO TPPAYMENTS VALUES (10010,8240224, 'CASH', 439.99, '2020-03-08');

INSERT INTO TPPAYMENTS VALUES (10011,8240237, 'AMEX', 453.99, '2020-03-18');

INSERT INTO TPPAYMENTS VALUES (10012,8240222, 'CASH', 400.59, '2020-03-28');

INSERT INTO TPPAYMENTS VALUES (10013,8240241, 'CASH', 234.59, '2020-04-09');

INSERT INTO TPPAYMENTS VALUES (10014,8240248, 'MASTERCARD', 149.59, '2020-04-10');

INSERT INTO TPPAYMENTS VALUES (10015,8240222, 'VISA', 229.76, '2020-04-12');

INSERT INTO TPPAYMENTS VALUES (10015,8240249, 'CASH', 129.64, '2020-04-20');

INSERT INTO TPPAYMENTS VALUES (10016,8240224, 'CASH', 529.74, '2020-04-25');

INSERT INTO TPPAYMENTS VALUES (10017,8240222, 'VISA', 829.60, '2020-04-27');

INSERT INTO TPPAYMENTS VALUES (10018,8240241, 'CASH', 229.81, '2020-05-03');

INSERT INTO TPPAYMENTS VALUES (10019,8240222, 'VISA', 262.65, '2020-05-10');

INSERT INTO TPPAYMENTS VALUES (10020,8240229, 'CASH', 259.56, '2020-05-15');

INSERT INTO TPPAYMENTS VALUES (10021,8240224, 'CASH', 251.56, '2020-05-24');

INSERT INTO TPPAYMENTS VALUES (10022,8240224, 'DISCOVER', 439.99, '2020-06-08');

INSERT INTO TPPAYMENTS VALUES (10023,8240237, 'AMEX', 453.99, '2020-06-17');

INSERT INTO TPPAYMENTS VALUES (10024,8240222, 'CASH', 149.59, '2020-06-23');

INSERT INTO TPPAYMENTS VALUES (10025,8240241, 'VISA', 234.59, '2020-07-09');

INSERT INTO TPPAYMENTS VALUES (10026,8240222, 'MASTERCARD', 299.89, '2020-07-21');

INSERT INTO TPPAYMENTS VALUES (10027,8240237, 'DISCOVER', 669.29, '2020-07-25');

INSERT INTO TPPAYMENTS VALUES (10028,8240226, 'VISA', 249.89, '2020-08-02');

INSERT INTO TPPAYMENTS VALUES (10029,8240229, 'CASH', 539.79, '2020-08-12');

INSERT INTO TPPAYMENTS VALUES (10030,8240222, 'DISCOVER', 262.65, '2020-08-15');

INSERT INTO TPPAYMENTS VALUES (10031,8240229, 'CASH', 258.56, '2020-08-18');

INSERT INTO TPPAYMENTS VALUES (10032,8240224, 'MASTERCARD', 500.56, '2020-08-20');

INSERT INTO TPPAYMENTS VALUES (10033,8240226, 'CASH', 299.89, '2020-09-14');

INSERT INTO TPPAYMENTS VALUES (10034,8240229, 'CASH', 599.19, '2020-09-19');

INSERT INTO TPPAYMENTS VALUES (10035,8240226, 'VISA', 229.89, '2020-10-02');

INSERT INTO TPPAYMENTS VALUES (10036,8240237, 'MASTERCARD', 399.99, '2020-10-05');

INSERT INTO TPPAYMENTS VALUES (10037,8240224, 'CASH', 109.00, '2020-10-27');

INSERT INTO TPPAYMENTS VALUES (10038,8240241, 'CASH', 234.59, '2020-11-11');

INSERT INTO TPPAYMENTS VALUES (10039,8240248, 'CASH', 149.59, '2020-11-17');

INSERT INTO TPPAYMENTS VALUES (10040,8240222, 'AMEX', 229.76, '2020-11-22');

INSERT INTO TPPAYMENTS VALUES (10041,8240249, 'AMEX', 129.64, '2020-12-06');

INSERT INTO TPPAYMENTS VALUES (10042,8240224, 'VISA', 529.74, '2020-12-12');

INSERT INTO TPPAYMENTS VALUES (10043,8240222, 'CASH', 829.60, '2020-12-17');

INSERT INTO TPPAYMENTS VALUES (10044,8240241, 'MASTERCARD', 229.81, '2020-12-26');

INSERT INTO TPINVOICE VALUES(200001,50009,8240231,1660,83349,"2020-1-1","AUSTIN","TEXAS","SOUTH",2);

INSERT INTO TPINVOICE VALUES(200002,50087,8240232,1682,83385,"2020-1-9","OAKLAND","CALIFORNIA","WEST",1);

INSERT INTO TPINVOICE VALUES(200003,50035,8240226,1660,83359,"2020-1-23","PALO ALTO","CALIFORNIA","WEST",1);

INSERT INTO TPINVOICE VALUES(200004,50052,8240246,1666,83366,"2020-1-25","HONOLULU","HAWAII","WEST",1);

INSERT INTO TPINVOICE VALUES(200005,50005,8240237,1682,83366,"2020-2-9","ARLINGTON","VIRGINIA","SOUTH",5);

INSERT INTO TPINVOICE VALUES(200006,50029,8240239,1660,83403,"2020-2-13","HOUSTON","TEXAS","SOUTH",4);

INSERT INTO TPINVOICE VALUES(200007,50011,8240233,1662,83403,"2020-2-22","NEW YORK","NEW YORK","NORTHEAST",3);

INSERT INTO TPINVOICE VALUES(200008,50068,8240238,1669,83312,"2020-3-8","NEW YORK","NEW YORK","NORTHEAST",4);

INSERT INTO TPINVOICE VALUES(200009,50009,8240244,1660,83378,"2020-3-11","PROVIDENCE","RHODE ISLAND","NORTHEAST",1);

INSERT INTO TPINVOICE VALUES(200010,50056,8240223,1662,83398,"2020-3-28","ATLANTA","GEORGIA","SOUTH",5);

INSERT INTO TPINVOICE VALUES(200011,50049,8240239,1661,83403,"2020-4-4","IRVINE","CALIFORNIA","WEST",1);

INSERT INTO TPINVOICE VALUES(200012,50038,8240234,1682,83312,"2020-4-20","ISELIN","NEW JERSEY","NORTHEAST",2);

INSERT INTO TPINVOICE VALUES(200013,50028,8240230,1689,83314,"2020-4-30","NEWARK","NEW JERSEY","NORTHEAST",3);

INSERT INTO TPINVOICE VALUES(200014,50031,8240222,1672,83366,"2020-5-5","ANN ARBOR","MICHIGAN","MIDWEST",4);

INSERT INTO TPINVOICE VALUES(200015,50009,8240248,1669,83308,"2020-5-8","SAN JOSE","CALIFORNIA","WEST",5);

INSERT INTO TPINVOICE VALUES(200016,50049,8240246,1661,83321,"2020-5-13","WILMINGTON","DELAWARE","SOUTH",3);

INSERT INTO TPINVOICE VALUES(200017,50068,8240230,1660,83332,"2020-5-24","HOBOKEN","NEW JERSEY","NORTHEAST",3);

INSERT INTO TPINVOICE VALUES(200018,50052,8240246,1669,83321,"2020-6-2","LOS ANGELES","CALIFORNIA","WEST",4);

INSERT INTO TPINVOICE VALUES(200019,50024,8240238,1672,83366,"2020-7-17","RICHMOND","VIRGINIA","SOUTH",4);

INSERT INTO TPINVOICE VALUES(200020,50067,8240235,1689,83347,"2020-7-19","DISTRICT OF COLUMBIA","DISTRICT OF COLUMBIA","SOUTH",4);

INSERT INTO TPINVOICE VALUES(200021,50028,8240232,1682,83312,"2020-7-19","OAKLAND","CALIFORNIA","WEST",4);

INSERT INTO TPINVOICE VALUES(200022,50031,8240232,1689,83403,"2020-7-20","SAN FRANCISCO","CALIFORNIA","WEST",3);

INSERT INTO TPINVOICE VALUES(200023,50030,8240245,1666,83371,"2020-7-24","MIAMI","FLORIDA","SOUTH",2);

INSERT INTO TPINVOICE VALUES(200024,50049,8240227,1666,83314,"2020-8-1","BERKELEY","CALIFORNIA","WEST",1);

INSERT INTO TPINVOICE VALUES(200025,50038,8240243,1661,83318,"2020-8-4","SEATTLE","WASHINGTON","WEST",5);

INSERT INTO TPINVOICE VALUES(200026,50068,8240246,1682,83382,"2020-8-13","CONCORD","NEW HAMPSHIRE","NORTHEAST",3);

INSERT INTO TPINVOICE VALUES(200027,50011,8240224,1660,83372,"2020-8-20","SALT LAKE CITY","UTAH","WEST",3);

INSERT INTO TPINVOICE VALUES(200028,50056,8240240,1661,83382,"2020-8-31","ORLANDO","FLORIDA","SOUTH",2);

INSERT INTO TPINVOICE VALUES(200029,50047,8240235,1682,83403,"2020-9-24","ISELIN","NEW JERSEY","NORTHEAST",2);

INSERT INTO TPINVOICE VALUES(200030,50005,8240227,1667,83385,"2020-9-26","PHILADELPHIA","PENNSYLVANIA","NORTHEAST",1);

INSERT INTO TPINVOICE VALUES(200031,50028,8240224,1661,83332,"2020-10-4","DISTRICT OF COLUMBIA","DISTRICT OF COLUMBIA","SOUTH",2);

INSERT INTO TPINVOICE VALUES(200032,50047,8240242,1660,83372,"2020-11-25","SAN DIEGO","CALIFORNIA","WEST",4);

INSERT INTO TPINVOICE VALUES(200033,50047,8240239,1676,83371,"2020-11-25","RIVERSIDE","CALIFORNIA","WEST",4);

INSERT INTO TPINVOICE VALUES(200034,50031,8240237,1660,83314,"2020-11-26","NEW YORK","NEW YORK","NORTHEAST",3);

INSERT INTO TPINVOICE VALUES(200035,50009,8240244,1660,83359,"2020-11-26","ALBANY","NEW YORK","NORTHEAST",1);

INSERT INTO TPINVOICE VALUES(200036,50052,8240238,1682,83366,"2020-11-26","CLEVELAND","OHIO","MIDWEST",2);

INSERT INTO TPINVOICE VALUES(200037,50052,8240247,1689,83308,"2020-11-26","ROCHESTER","NEW YORK","NORTHEAST",2);

INSERT INTO TPINVOICE VALUES(200038,50067,8240242,1662,83312,"2020-12-27","MOUNTAIN VIEW","CALIFORNIA","WEST",5);

INSERT INTO TPINVOICE VALUES(200039,50067,8240240,1682,83308,"2020-12-28","DALLAS","TEXAS","SOUTH",2);

INSERT INTO TPINVOICE VALUES(200040,50049,8240237,1676,83321,"2020-12-30","NEW YORK","NEW YORK","NORTHEAST",5);

INSERT INTO TPINVOICE VALUES(200041,50049,8240225,1669,83382,"2020-12-30","STAMFORD","CONNECTICUT","NORTHEAST",5);

INSERT INTO TPINVOICE VALUES(200042,50038,8240226,1669,83385,"2020-12-31","MOUNTAIN VIEW","CALIFORNIA","WEST",1);

INSERT INTO TPINVOICE VALUES(200043,50031,8240247,1682,83398,"2020-12-31","DALLAS","TEXAS","SOUTH",3);

INSERT INTO TPPRODUCT VALUES (50062,2001,103,1204,'graphics tablet',1399);

INSERT INTO TPPRODUCT VALUES (50014,2001,103,1202,'mechanical keyboard',399);

INSERT INTO TPPRODUCT VALUES (50031,2001,104,1204,'DSLR camera',1499);

INSERT INTO TPPRODUCT VALUES (50024,2011,103,1203,'powered sit to stand table',1799);

INSERT INTO TPPRODUCT VALUES (50005,2012,103,1204,'studio headphone',899);

INSERT INTO TPPRODUCT VALUES (50035,2012,101,1201,'commercial photocopier',2299);

INSERT INTO TPPRODUCT VALUES (50068,2012,101,1204,'85inch Screen 4K',1315);

INSERT INTO TPPRODUCT VALUES (50052,2012,104,1204,'4K projector',2089);

INSERT INTO TPPRODUCT VALUES (50030,2019,103,1205,'Purifier Humidify+Cool Formaldehyde',1099);

INSERT INTO TPPRODUCT VALUES (50028,2019,102,1203,'massage chair',3885);

INSERT INTO TPPRODUCT VALUES (50100,2019,104,1201,'file cabinet trolley aluminum',999);

INSERT INTO TPPRODUCT VALUES (50029,2019,104,1205,'full auto coffee machine',1278);

INSERT INTO TPPRODUCT VALUES (50049,2019,101,1204,'ergonomics chair',1178);

INSERT INTO TPPRODUCT VALUES (50047,2019,103,1203,'group sofa',2025);

INSERT INTO TPPRODUCT VALUES (50038,2019,104,1203,'lounge chair',999);

INSERT INTO TPPRODUCT VALUES (50067,2098,104,1206,'512GB SD card',159);

INSERT INTO TPPRODUCT VALUES (50056,2098,102,1202,'4TB Server',2499);

INSERT INTO TPPRODUCT VALUES (50087,2098,102,1201,'Workstation',1899);

INSERT INTO TPPRODUCT VALUES (50011,2098,103,1202,'QHD 24inch Display',599);

INSERT INTO TPPRODUCT VALUES (50009,2098,103,1204,'HD stereo system',699);

**Part 5:**

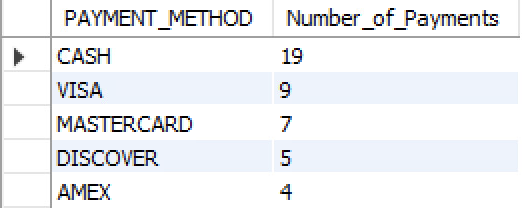
1. **Write a query to display the number of payments in each category. Sort by the payments total in descending order. Use an alias wherever required.**

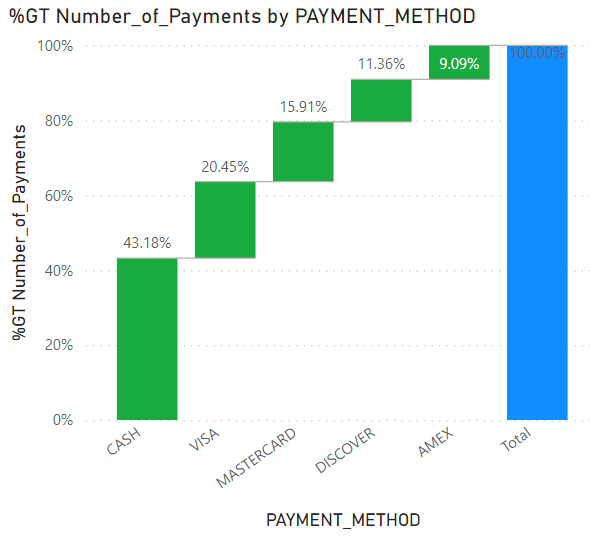
SELECT PAYMENT\_METHOD, COUNT(\*) AS NUMBER\_OF\_PAYMENTS

FROM TPPAYMENTS

GROUP BY PAYMENT\_METHOD

ORDER BY NUMBER\_OF\_PAYMENTS DESC;

****

****

**Majority of our sales are paid by credit cards versus cash, with 57% of the transactions being in credit. It is consistent with the fact that most people prefer paying by cards now.**

**2. How many sales have been made in New York City by the sales channel Best Buy. Use an alias wherever required.**

SELECT CITY, CHANNEL\_NAME ,COUNT(INVOICE\_ID) AS SALES

FROM TPINVOICE I, TPSALES\_CHANNEL SC

WHERE I.S\_CHANNEL\_ID=SC.S\_CHANNEL\_ID

AND CITY = 'NEW YORK'

AND CHANNEL\_NAME='BESTBUY'

GROUP BY CITY, CHANNEL\_NAME;



**There is only one sale that has been made in New York by Best Buy. This is a channel we were looking to capitalize and expand since we believe there is a big market for our products in New York. The number of sales is unsatisfactory as of now.**

**3. Which employee made the most sales. Sort by the sales total in descending order. Use an alias wherever required.**

SELECT EMPLOYEE\_NAME AS SALES\_REP, SUM(QUANTITY\*PRODUCT\_PRICE) AS SALES

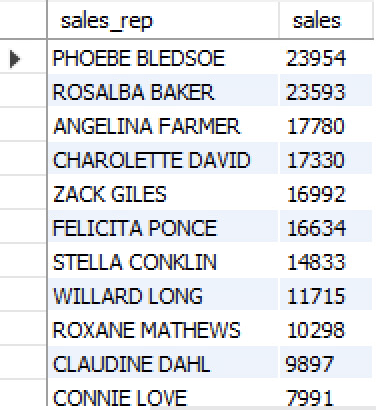
FROM TPEMPLOYEE E, TPINVOICE I, TPPRODUCT P

WHERE E.EMPLOYEE\_ID=I.SALES\_REP\_ID

AND I.PRODUCT\_ID=P.PRODUCT\_ID

GROUP BY SALES\_REP

ORDER BY SALES DESC;



**We wanted to rank our employees based on sales and find our top salesmen, since our employee’s performance is measured in terms of total sales.**

**4. Write a query to display the biggest supplier. Use an alias wherever required.**

SELECT SUPPLIER\_NAME , COUNT(PRODUCT\_NAME) AS NUMBER\_OF\_PRODUCTS

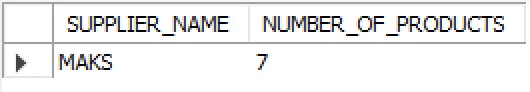
FROM TPSUPPLIER S, TPPRODUCT P

WHERE S.SUPPLIER\_ID = P.SUPPLIER\_ID

GROUP BY SUPPLIER\_NAME

ORDER BY NUMBER\_OF\_PRODUCTS DESC

LIMIT 1;

****

**We wanted to find out who our biggest supplier was. We found out in terms of the total number of products they supply to us. We can now decide whether we should consider building a long term relationship with MAKS or sourcing more products from other suppliers and make our supplier market more competitive. Either approach can significantly impact our next contract negotiation with the supplier.**

**5. Write a query to display a product id and the number of products that are sold. Sort the output by the unit\_sales in descending order. Use an alias wherever required.**

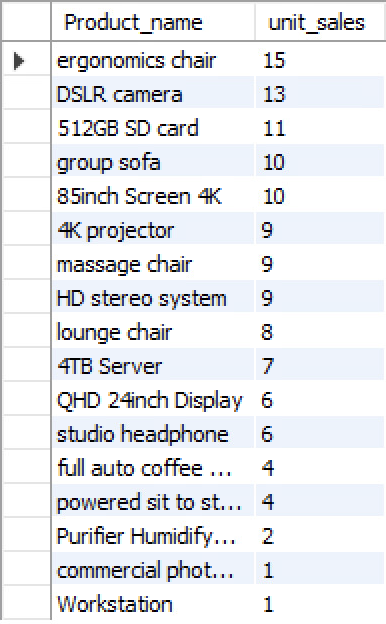
SELECT PRODUCT\_NAME, SUM(QUANTITY) AS UNIT\_SALES

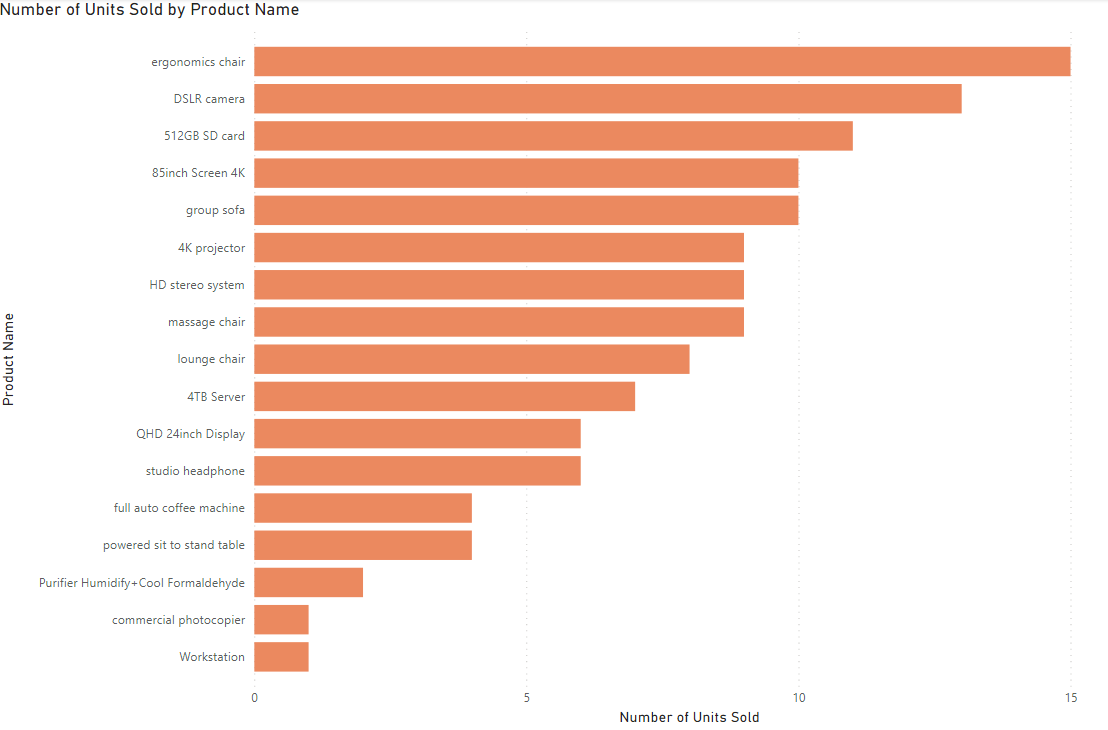
FROM TPPRODUCT P, TPINVOICE I

WHERE P.PRODUCT\_ID=I.PRODUCT\_ID

GROUP BY PRODUCT\_NAME

ORDER BY UNIT\_SALES DESC;





**‘Ergonomics chair’ is our best selling product. We have a good unit sales distribution between our best selling products which is critical for a business’ long term positive cash flow.**

**6. Display sales amount, number of sales over time by month. Sort by monthly sales in descending order. Use an alias wherever required.**

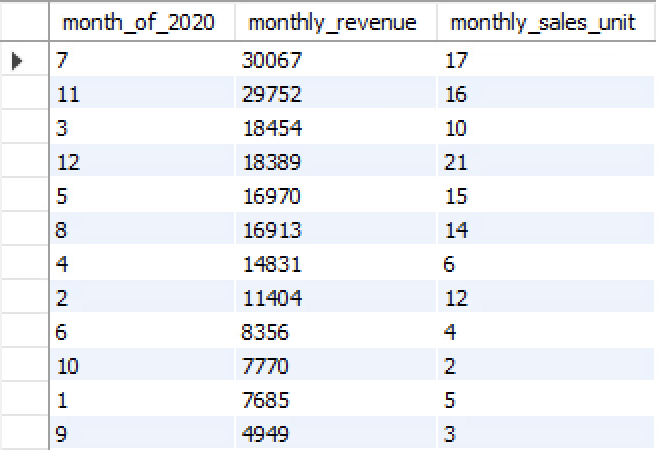
SELECT MONTH(INV\_DATE) AS MONTH\_OF\_2020, SUM(QUANTITY\*PRODUCT\_PRICE) AS MONTHLY\_REVENUE, SUM(QUANTITY) AS MONTHLY\_SALES\_UNIT

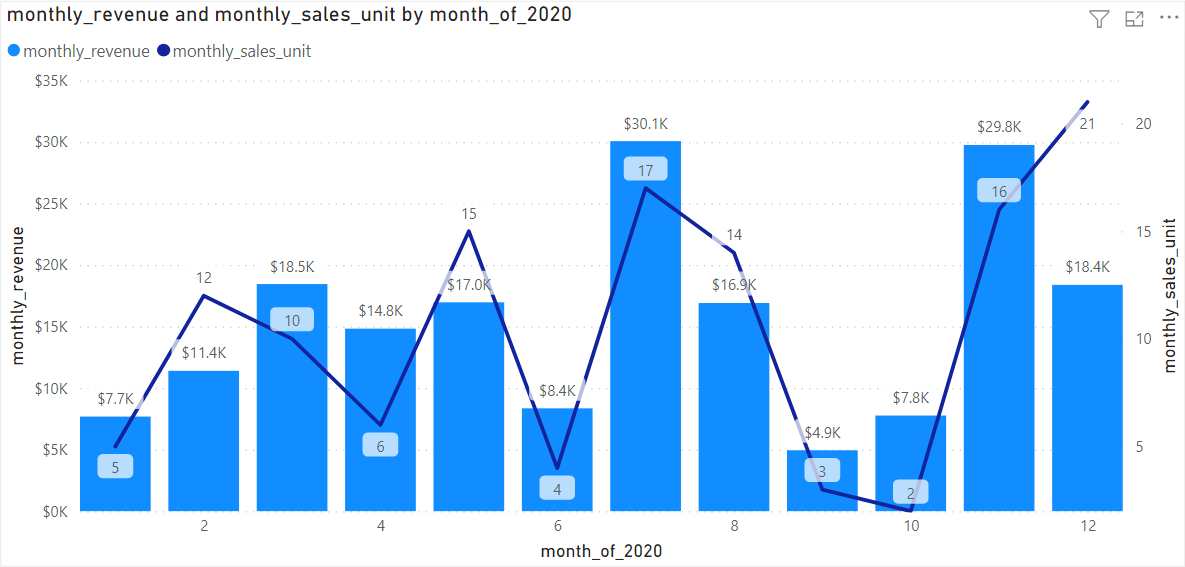
FROM TPINVOICE TI, TPPRODUCT TP

WHERE TI.PRODUCT\_ID=TP.PRODUCT\_ID

GROUP BY MONTH\_OF\_2020

ORDER BY MONTHLY\_REVENUE DESC;

****

****

**Months of July, November and December has the highest sales. This is consistent with the typical sale trends in the US, which is the summer sales and then the holiday sales. Month of March could be an outlier for this year, we need to see data of other years to confirm that claim though.**

**7. Display sales by each region. Sort the results by sales in ascending order. Use an alias wherever required.**

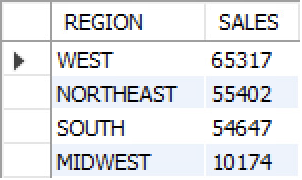
SELECT REGION, SUM(QUANTITY\* PRODUCT\_PRICE) AS SALES

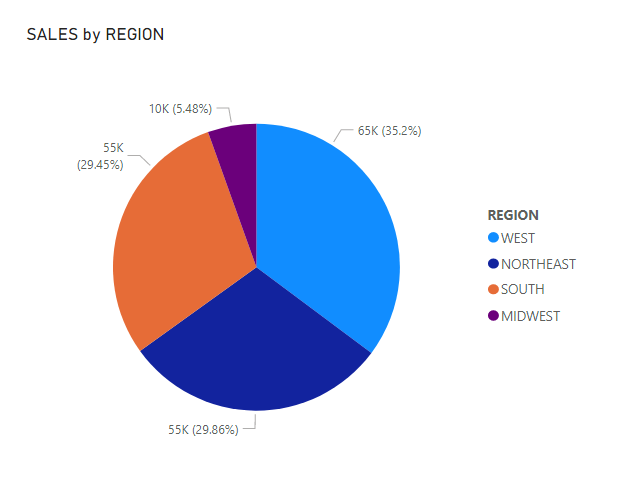
FROM TPPRODUCT P, TPINVOICE I

WHERE P.PRODUCT\_ID = I.PRODUCT\_ID

GROUP BY REGION

ORDER BY SALES DESC;





**We wanted to see how each region is performing in terms of sales. Although three of the four regions are competing very well against each other at the top, Midwest region is far behind the rest. We need to analyze the market scope in that region and decide whether we should continue or shut down selling in the Midwest region.**

**8. How many customers are spending more than $2000 in the west region.**

SELECT COUNT(\*) AS NUMCUS\_ABOVE\_AVERAGE

FROM

(SELECT CUSTOMER\_NAME, SUM(QUANTITY\*PRODUCT\_PRICE) AS REVENUE

FROM TPINVOICE I, TPCUSTOMER C, TPPRODUCT P

WHERE I.CUSTOMER\_ID=C.CUSTOMER\_ID

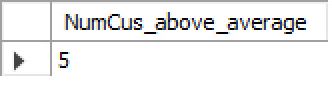
AND I.PRODUCT\_ID=P.PRODUCT\_ID

AND REGION="WEST"

GROUP BY CUSTOMER\_NAME

HAVING REVENUE > AVG(QUANTITY\*PRODUCT\_PRICE)

ORDER BY REVENUE DESC) AS PURCHASE;



**From our findings that the West region has the highest sales amongst all the regions, we wanted to see how many customers they have who have purchased above the average customer spending. This gives us the opportunity to build on a loyal customer base in our strongest region.**

**9. How many sales have been made by each sales channel. Sort the results by sales in ascending order. Use an alias wherever required.**

SELECT CHANNEL\_NAME AS CHANNELS, SUM(QUANTITY) AS UNIT\_SALES

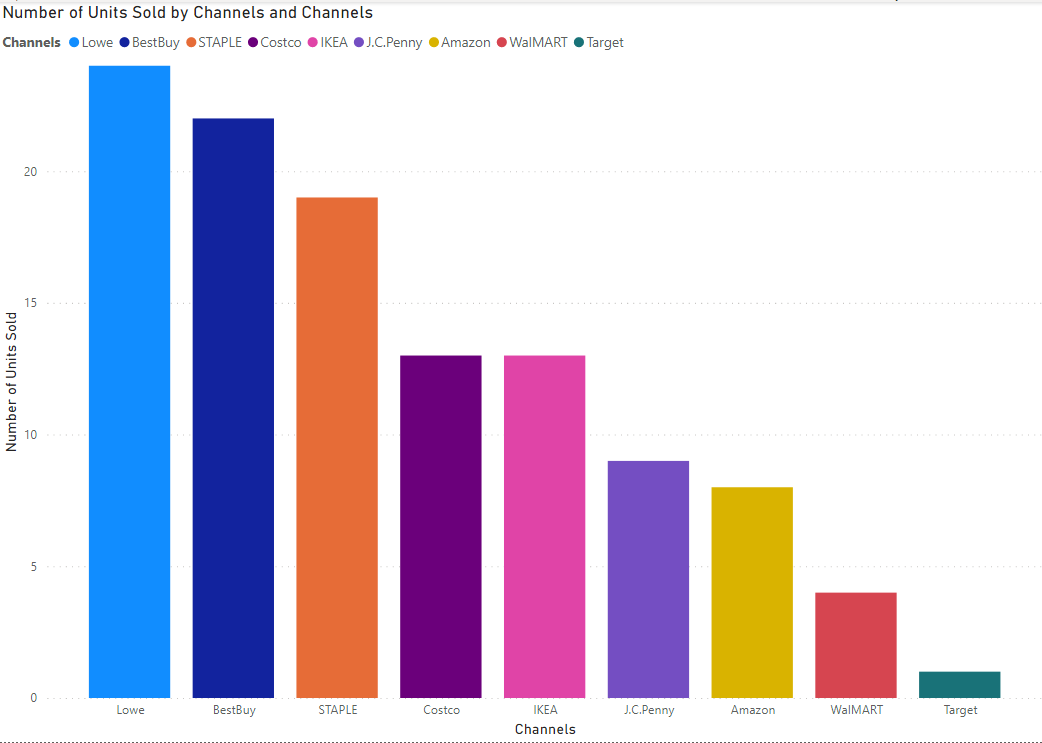
FROM TPINVOICE TI,TPPRODUCT TP, TPSALES\_CHANNEL TS

WHERE TI.PRODUCT\_ID=TP.PRODUCT\_ID AND TI.S\_CHANNEL\_ID=TS.S\_CHANNEL\_ID

GROUP BY CHANNELS

ORDER BY UNIT\_SALES ASC;

****

****

**In this query, we can find out the sales distribution of our products through all our sales channels. We can use this data to determine future budget and product allocation.**

**10. Write a query to display the average product price by product type. Sort the results by sales in descending order. Use an alias wherever required.**

SELECT PRODUCT\_TYPE, AVG(PRODUCT\_PRICE) AS AVERAGE\_PRICE, COUNT(P.PRODUCT\_ID) AS QUANITY\_SOLD

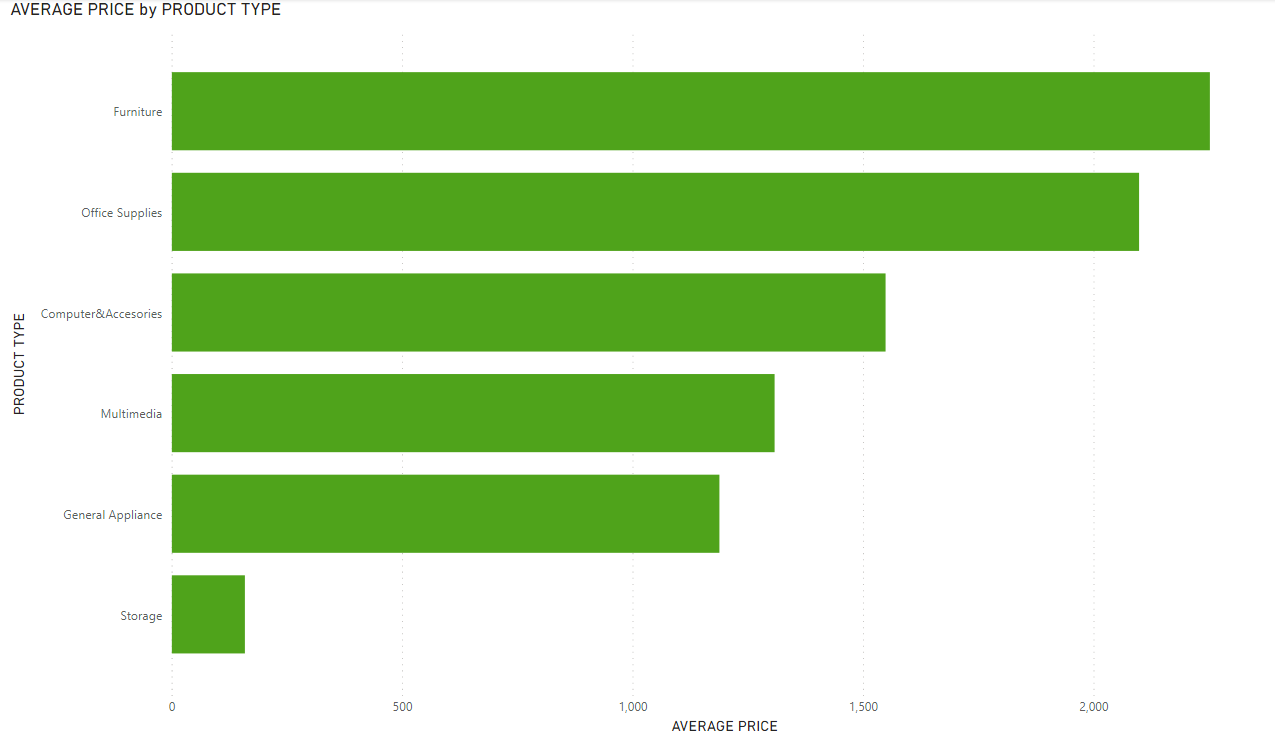
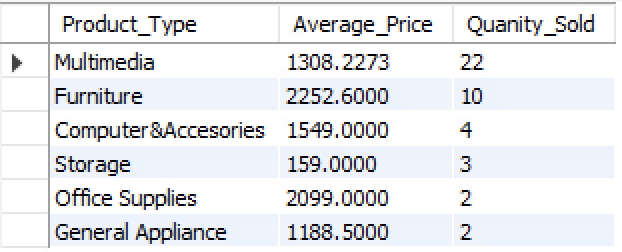
FROM TPPRODUCT P, TPPRODUCT\_TYPE PT, TPINVOICE IV

WHERE P.P\_TYPE\_ID = PT.P\_TYPE\_ID

AND IV.PRODUCT\_ID = P.PRODUCT\_ID

GROUP BY PRODUCT\_TYPE

ORDER BY QUANITY\_SOLD DESC;

****

**We wanted to see the unit sales distribution of our different product types and determine which product types are not doing well. We need to consider either adding more products to those categories, or increasing advertisement/marketing for the current product lineup, or liquidating the product type completely.**

**11. Write a query to display Top 5 states that have made the highest sales. Sort the results by sales in descending order. Use an alias wherever required.**

SELECT STATE, SUM(QUANTITY\*PRODUCT\_PRICE) AS TOTAL\_SALES

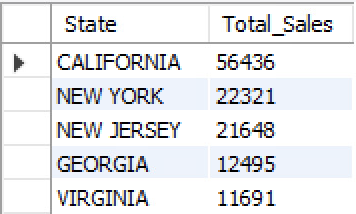
FROM TPINVOICE IV, TPPRODUCT PD

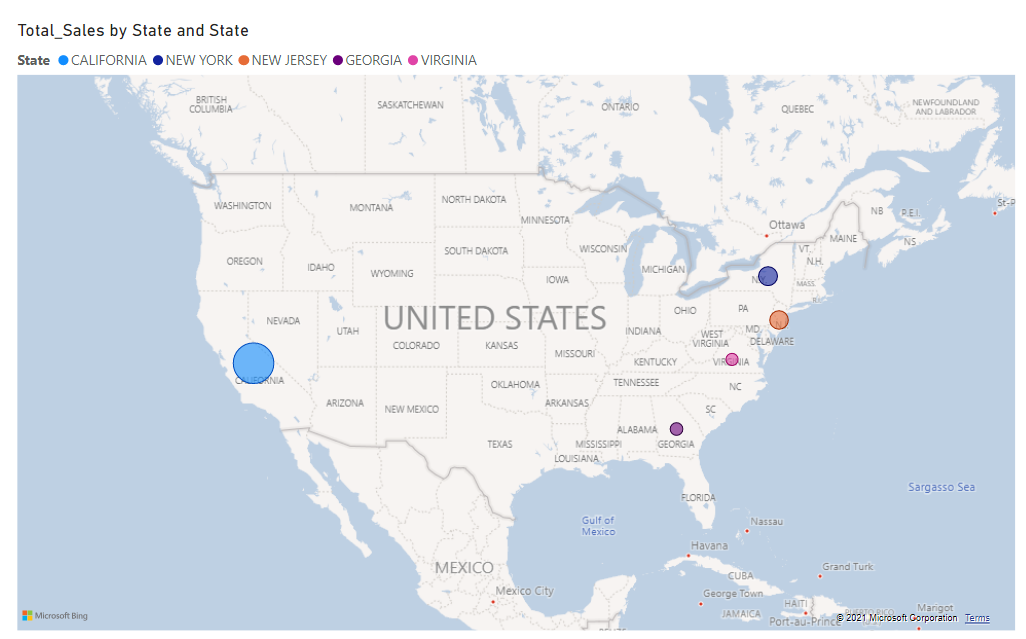
WHERE IV.PRODUCT\_ID = PD.PRODUCT\_ID

GROUP BY STATE

ORDER BY TOTAL\_SALES DESC

LIMIT 5;





**We wanted to see our top 5 states which consist of most sales of our products. The sales are in line with the total population of these states. Using this information, we could determine launching one or two specialized products to states in a particular region which might break the current situation and make sales breakthrough even with less population, since we cannot increase the market size, but we can capture new market segments.**

**12. Write a query to display top 5 customers who have made the highest purchase that are from California. Sort the results by sales in descending order. Use an alias wherever required.**

SELECT CUSTOMER\_NAME AS CUSTOMER, STATE, SUM(QUANTITY\*PRODUCT\_PRICE) AS PURCHASE

FROM TPINVOICE TI, TPCUSTOMER TC, TPPRODUCT TP

WHERE TI.CUSTOMER\_ID=TC.CUSTOMER\_ID

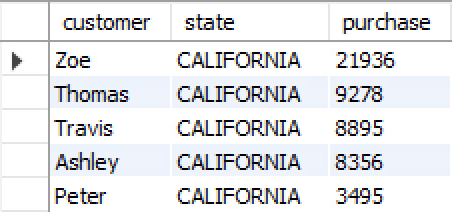
AND TI.PRODUCT\_ID=TP.PRODUCT\_ID

AND STATE='CALIFORNIA'

GROUP BY CUSTOMER

ORDER BY PURCHASE DESC

LIMIT 5;

****

**We can run this query to find out who our biggest and most loyal customers are. We are planning to launch a customer membership system in the future so this data will help us determine the point system scale that would best benefit the business, and also our customers.**

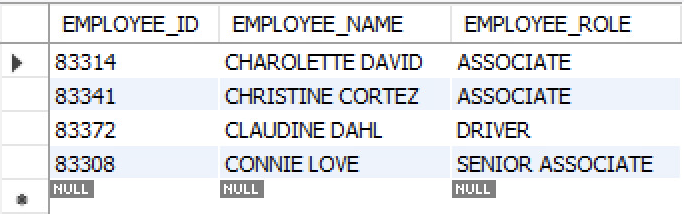
**13. Write a query to display all the employees’ first names that start with “C”. Sort the results by sales in alphabetical order. Use an alias wherever required.**

SELECT EMPLOYEE\_ID, EMPLOYEE\_NAME, EMPLOYEE\_ROLE

FROM TPEMPLOYEE

WHERE EMPLOYEE\_NAME LIKE 'C%'

ORDER BY EMPLOYEE\_NAME ASC;



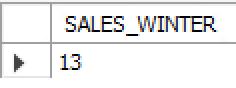
**This query was made with the purpose of being quickly able to identify a particular associate if needed. For example, customers might often refer to an employee but will not remember their name completely. This query can be adjusted accordingly and will give a list of employees that might help our customers identify the employee they are referring to.**

**14. Write a query to display the number of sales made during the winter season. Use an alias wherever required.**

SELECT COUNT(INVOICE\_ID)/COUNT(\*) AS SALES\_WINTER

FROM TPINVOICE

WHERE INV\_DATE BETWEEN '2020-10-1' AND '2021-2-1';



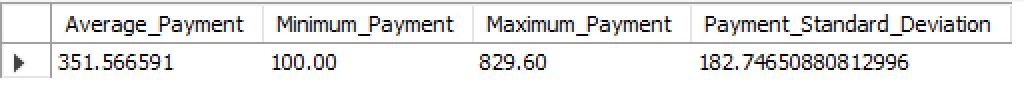
**We wanted to see how many items we sold during the holiday season. We wanted to use this as a benchmark and build on this in the future since it is critical to make full use of the holiday season to increase sales.**

**15. Write a query to display descriptive statistics (like average, minimum, maximum, standard deviation) for payments**

SELECT AVG(PAYMENT\_AMOUNT) AS AVERAGE\_PAYMENT, MIN(PAYMENT\_AMOUNT) AS MINIMUM\_PAYMENT, MAX(PAYMENT\_AMOUNT) AS MAXIMUM\_PAYMENT,

STD(PAYMENT\_AMOUNT) AS PAYMENT\_STANDARD\_DEVIATION

FROM TPPAYMENTS;



**We wanted to have the basic statistics calculated for customer payments. We could compare these statistics with our product price statistics to get a clearer picture of our customer behaviors. The big range and a standard deviation of $182 makes sense with our product prices being quite different from each other. This gives us a baseline to work and build upon on a daily basis business operation. However, in order to make bigger strategic decisions, we would also need to incorporate the purchase quantity between customers since that can also factor into these statistics.**