**Team BRAMS**

**Members:** Bryan Vu, Rachel Pai, Autumn Nguyen,   
Marjorie Balaoro and Saurav Chhapawala  
CECS 323 Term Project: Triggers

**TRIGGER #1**

/\* This is an Before Insert trigger is to ensure that the information for the CustomerAccounts matches with the type of Customer the account is (Private or Corporation).  
If the account is Private: Email attribute is filled.

If the account is Corporation: CorporationName and DepartmentName is filled.

If the insert statement does not agree to the above conditions, an error will occur and the statement will not go through.\*/

DROP TRIGGER IF EXISTS `cecs323sec07bg04`.`CustomerAccounts\_BEFORE\_INSERT`;

DELIMITER $$

USE `cecs323sec07bg04`$$

CREATE DEFINER=`cecs323sec07s36`@`10.%` TRIGGER `cecs323sec07bg04`.`CustomerAccounts\_BEFORE\_INSERT` BEFORE INSERT ON `CustomerAccounts` FOR EACH ROW

BEGIN

DECLARE msg VARCHAR(128);

IF (NEW.CustomerType = 'Private') THEN

IF (NEW.Email = NULL) OR (NEW.CorporationName IS NOT NULL) OR (NEW.DepartmentName IS NOT NULL) THEN

SET msg = concat('CustomerAccountsTriggerError: CustomerType is Private but Email was not filled or Corporation Information was filled.');

signal sqlstate '45000' set message\_text = msg;

END IF;

ELSEIF (NEW.CustomerType = 'Corporation') THEN

IF (NEW.CorporationName = NULL) OR (NEW.DepartmentName = NULL) OR (NEW.Email IS NOT NULL) THEN

SET msg = concat('CustomerAccountsTriggerError: CustomerType is Corporation but Corporation Information was not filled or Email was filled.');

signal sqlstate '45000' set message\_text = msg;

END IF;

END IF;

END$$

DELIMITER ;

**SAMPLE OUTPUT #1**

**SQL CODE (Inserting Wrong Info for Private):** INSERT INTO CustomerAccounts (CustomerID, CustomerName, PhysicalAddress, CustomerType, CorporationName, DepartmentName, Email, MimingsMoneyAmount)

VALUES (23, 'Bianca Glimmer', '437 Columbia Dr. Knoxville, TN 37918', 'Private', 'Samsung', 'Support', NULL, 0);

**OUTPUT:** Error Code: 1644. CustomerAccountsTriggerError: CustomerType is Private but Email was not filled or Corporation Information was filled. 0.015 sec

**SQL CODE (Inserting Wrong Info for Corporation):** INSERT INTO CustomerAccounts (CustomerID, CustomerName, PhysicalAddress, CustomerType, CorporationName, DepartmentName, Email, MimingsMoneyAmount)

VALUES

(24, 'Hugh Oink', '104 Church Rd. Smyrna, GA 30080', 'Corporation', NULL, NULL, 'hughoink@gmail.com', 0);

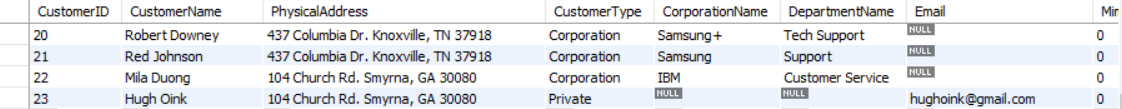
**OUTPUT:** Error Code: 1644. CustomerAccountsTriggerError: CustomerType is Corporation but Corporation Information was not filled or Email was filled. 0.109 sec

**SQL CODE (Inserting Right Info for Private):** INSERT INTO CustomerAccounts (CustomerID, CustomerName, PhysicalAddress, CustomerType, CorporationName, DepartmentName, Email, MimingsMoneyAmount)

VALUES

(23, 'Hugh Oink', '104 Church Rd. Smyrna, GA 30080', 'Private', NULL, NULL, 'hughoink@gmail.com', 0);

**OUTPUT:** 1 row(s) affected 0.016 sec



**TRIGGER #2**

/\* This is an Before Update trigger is to ensure that the information for the CustomerAccounts matches with the type of Customer the account is (Private or Corporation).  
If the account is Private: Email attribute is filled.

If the account is Corporation: CorporationName and DepartmentName is filled.

If the update statement does not agree to the above conditions, an error will occur and the statement will not go through.\*/

DROP TRIGGER IF EXISTS `cecs323sec07bg04`.`CustomerAccounts\_BEFORE\_UPDATE`;

DELIMITER $$

USE `cecs323sec07bg04`$$

CREATE DEFINER=`cecs323sec07s36`@`10.%` TRIGGER `cecs323sec07bg04`.`CustomerAccounts\_BEFORE\_UPDATE` BEFORE UPDATE ON `CustomerAccounts` FOR EACH ROW

BEGIN

DECLARE msg VARCHAR(128);

IF (NEW.CustomerType = 'Private') THEN

IF (NEW.CorporationName IS NOT NULL) OR (NEW.DepartmentName IS NOT NULL) THEN

SET msg = concat('CustomerAccountsTriggerError: CustomerType is Private but Email was not filled or Corporation Information was filled.');

signal sqlstate '45000' set message\_text = msg;

END IF;

ELSEIF (NEW.CustomerType = 'Corporation') THEN

IF (NEW.CorporationName = NULL) OR (NEW.DepartmentName = NULL) OR (NEW.Email IS NOT NULL) THEN

SET msg = concat('CustomerAccountsTriggerError: CustomerType is Corporation but Corporation Information was not filled or Email was filled.');

signal sqlstate '45000' set message\_text = msg;

END IF;

END IF;

END$$

DELIMITER ;

**SAMPLE OUTPUT #2**

**SQL CODE (Updating Wrong Info for Private):** UPDATE CustomerAccounts

SET CustomerName = 'Autumn Moo', CorporationName= 'Palo Alto Networks'

WHERE CustomerID = 1;

**OUTPUT:** Error Code: 1644. CustomerAccountsTriggerError: CustomerType is Private but Email was not filled or Corporation Information was filled. 0.015 sec

**SQL CODE (Updating Wrong Info for Corporation):** UPDATE CustomerAccounts

SET CorporationName= 'Samsung+', Email= 'samsung@support.com'

WHERE CustomerID = 20;

**OUTPUT:** Error Code: 1644. CustomerAccountsTriggerError: CustomerType is Corporation but Corporation Information was not filled or Email was filled. 0.031 sec

**SQL CODE (Updating Right Info for Corporation):** UPDATE CustomerAccounts

SET CorporationName= 'Samsung+', DepartmentName= 'Tech Support'

WHERE CustomerID = 20;

**OUTPUT:** 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0 0.141 sec

**TRIGGER #3**

/\* This is an Before Insert trigger that calculates an added gratuity charge to the EatInOrder if the party size of the order is greater than or equal to a size of 8.\*/

DROP TRIGGER IF EXISTS `cecs323sec07bg04`.`EATINORDER\_BEFORE\_INSERT`;

DELIMITER $$

USE `cecs323sec07bg04`$$

CREATE DEFINER=`cecs323sec07s36`@`10.%` TRIGGER `cecs323sec07bg04`.`EATINORDER\_BEFORE\_INSERT` BEFORE INSERT ON `EATINORDER` FOR EACH ROW

BEGIN

IF ((SELECT PartySize FROM CUSTOMERORDER WHERE CUSTOMERORDER.ORDERID = NEW.ORDERID) >= 8) THEN

SET NEW.GratuityCharge = (SELECT AmountTotal FROM CUSTOMERORDER WHERE CUSTOMERORDER.ORDERID = NEW.ORDERID)\*.20;

END IF;

END$$

DELIMITER ;

**SAMPLE OUTPUT #3**

**SQL CODE (Inserting new CustomerOrder with PartySize of 1):**

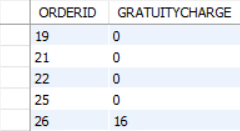
INSERT INTO CUSTOMERORDER (CUSTOMERID, CUSTOMERNAME, PARTYSIZE, ORDERID, ORDERDATE, ORDERTIME, AMOUNTTOTAL)

VALUES (1, 'Autumn Nguyen', 1, 25, '2020-05-06', '09:00:00', 10);

INSERT INTO EATINORDER (ORDERID, GRATUITYCHARGE)

VALUES (25, 0);

**OUTPUT: OrderID = 25**



**SQL CODE (Inserting new CustomerOrder with PartySize of 8):**

INSERT INTO PARTY (CUSTOMERID, CUSTOMERNAME, PARTYSIZE)

VALUES (2, 'Bryan Vu', 8);

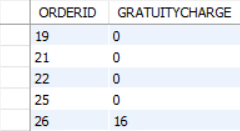
INSERT INTO CUSTOMERORDER (CUSTOMERID, CUSTOMERNAME, PARTYSIZE, ORDERID, ORDERDATE, ORDERTIME, AMOUNTTOTAL)

VALUES (2, 'Bryan Vu', 8, 26, '2020-05-06', '12:00:00', 80);

INSERT INTO EATINORDER (ORDERID, GRATUITYCHARGE)

VALUES (26, 0);

**OUTPUT: OrderID = 26**



**TRIGGER #4**

/\* This is an Before Insert trigger that determines if the total of the Order is above the $5 minimum charge. If it is, then the Order is inserted into the Card table successfully. If it doesn't meet the minimum, an error message will popup and the statement will not go through.\*/

DROP TRIGGER IF EXISTS `cecs323sec07bg04`.`CARD\_BEFORE\_INSERT`;

DELIMITER $$

USE `cecs323sec07bg04`$$

CREATE DEFINER=`cecs323sec07s36`@`10.%` TRIGGER `cecs323sec07bg04`.`CARD\_BEFORE\_INSERT` BEFORE INSERT ON `CARD` FOR EACH ROW

BEGIN

DECLARE msg VARCHAR(128);

IF ((SELECT AmountTotal FROM CUSTOMERORDER WHERE CUSTOMERORDER.ORDERID = NEW.ORDERID) < 5) THEN

SET msg = concat('CardError: Order did not meet the minimum charge of $5, please add this order using another payment option.');

signal sqlstate '45000' set message\_text = msg;

END IF;

END$$

DELIMITER ;

**SQL CODE (Inserting a new order that did not meet the $5 requirement):** INSERT INTO CUSTOMERORDER (CUSTOMERID, CUSTOMERNAME, PARTYSIZE, ORDERID, ORDERDATE, ORDERTIME, AMOUNTTOTAL)

VALUES (1, 'Autumn Nguyen', 1, 23, '2020-05-06', '09:00:00', 2);

INSERT INTO CARD (ORDERID, CARDTYPE)

VALUES (23, 'DebitCard');

**OUTPUT:** Error Code: 1644. CardError: Order did not meet the minimum charge of $5, please add this order using another payment option. 0.016 sec

**SQL CODE (Inserting a new order that did meet the $5 requirement):** INSERT INTO CUSTOMERORDER (CUSTOMERID, CUSTOMERNAME, PARTYSIZE, ORDERID, ORDERDATE, ORDERTIME, AMOUNTTOTAL)

VALUES (1, 'Autumn Nguyen', 1, 24, '2020-05-06', '09:00:00', 10);

INSERT INTO CARD (ORDERID, CARDTYPE)

VALUES (24, 'DebitCard');

**OUTPUT:** 1 row(s) affected 0.031 sec

