/\* 1 Select all customers who created account in month of june\*/

Select `First\_name`, `Last\_name`

From `CUSTOMERS`

WHERE `Created\_on` LIKE '%-06-%';

/\* 2 Insert details of new customer \*/

INSERT INTO `CUSTOMERS` (`Acc\_no`,`First\_name`,`Last\_name`,`DOB`,`Contact\_No`,`Address`,`Created\_on`,`Amount`,`ATM\_NO`,`PIN`,`Emp\_id`)

VALUES (1239799085, 'Akshay', 'Vaje', '1999-12-14', 9431556252, 'Virar', '2014-06-03', 214748, 45911254873610369, 3543, 'alex\_banker');

/\* 3 Delete the employee with employid 'alex\_banker' \*/

/\*

DELETE FROM `Employee`

WHERE Emp\_id='alex\_banker';

\*/

/\* 4 Update the details of customer with accout no 1239799085 \*/

Update `CUSTOMERS`

SET `DOB`= '2000-12-14'

WHERE `Acc\_no`=1239799085;

/\* 5 Order all customers according to their Date of Birth \*/

SELECT `DOB`

FROM `CUSTOMERS`

ORDER BY `DOB`

ASC;

/\* 6 Select all customer with amount greater than 10000 \*/

SELECT `First\_name`,`Last\_name`,`Acc\_no`

FROM `CUSTOMERS`

WHERE `Amount` > 10000;

/\* 7 Select all the transaction with amountpython greater than 45000 and transaction id =20 \*/

SELECT `Acc\_no1`, `Acc\_no2`

FROM `Transaction`

WHERE `Amount` > 45000 AND `Trans\_id`=20;

/\* 8 Retrieve all cutomers with differnt Date of Birth \*/

SELECT DISTINCT `DOB`

FROM `CUSTOMERS`;

/\* 9 select all customers with age greater than 60 and age less than 18 8\*/

SELECT `First\_name`,`Last\_name`,`Acc\_no`

FROM `CUSTOMERS`

WHERE `DOB` <='1961-04-17' AND `DOB`>='2011-04-17';

/\* 10 Selct transaction id of account with account no 12379129950 and account no 1237912995 \*/

SELECT `Trans\_id`

FROM `Transaction`

WHERE `Acc\_no1`=1237912995 AND `Acc\_no2`=12379129950;

/\* 11 Update password of employee with empid=ATM\_banker \*/

UPDATE `Employee`

SET `Password`='ajsfjsnis15485'

WHERE `Emp\_id`='ATM\_banker';

/\* 12 Order all the employee according to descending order of empid \*/

SELECT `Emp\_id`

FROM `Employee`

ORDER BY `Emp\_id`

DESC;

/\* 13 inner join customer account no with transaction id \*/

SELECT \*

FROM `CUSTOMERS`

INNER JOIN `Transaction`

ON `CUSTOMERS.Acc\_No` = `Transaction.Acc\_No1`;

/\* 14 Left Join customer account no with transaction id \*/

SELECT \*

FROM `CUSTOMERS`

LEFT OUTER JOIN Customer

ON `CUSTOMERS.Acc\_No` = `Transaction.Acc\_No1`;

/\* 15 Make sure account balance of employee is not less than zero \*/

SELECT \*

FROM `CUSTOMERS`

WHERE `Amount` IS NOT NULL;

/\* 16 Delete customer with account no=1238153062 \*/

/\*

DELETE FROM `CUSTOMERS`

WHERE `Acc\_no`=1238153062;

\*/

/\* 17 Insert details of new employee \*/

INSERT INTO `Employee` (`Emp\_id`, `Password`)

VALUES ('VR\_115', '3225016616611ajsksj');

/\* 18 Order all cutomers according to their pin \*/

SELECT `First\_name`,`Last\_name`,`Acc\_no`

FROM `CUSTOMERS`

ORDER BY `PIN`

ASC;

/\* 19 get customer with contatc no=7525992132 \*/

SELECT `First\_name`,`Last\_name`,`Acc\_no`

FROM `CUSTOMERS`

WHERE `Contact\_No`=7525992132;

/\* 20 Select everything from transaction table \*/

SELECT \* FROM `Transaction`;

/\* 21 Create new Account for new Customer \*/

INSERT INTO `CUSTOMERS` (`Acc\_no`,`First\_name`,`Last\_name`,`DOB`,`Contact\_No`,`Address`,`Created\_on`,`Amount`,`ATM\_NO`,`PIN`,`Emp\_id`)

VALUES (1248959085, 'Vaibhav', 'Ranshoor', '1999-08-27', 9431254622, 'Thane', '2018-07-25', 5000000, 459141256190320619, 3871, 'alex\_banker');

/\* 22 Order all transaction on basis of date \*/

SELECT `Acc\_no1` ,`Acc\_no2`

FROM `Transaction`

ORDER BY `Date`

ASC;

/\* 23 Get last name of customer with first name =Vaibhav \*/

SELECT `Last\_name`

FROM `CUSTOMERS`

WHERE `First\_name`='Vaibhav';

/\* 24 remove transaction with transaction id =15 \*/

DELETE FROM `Transaction`

WHERE `Trans\_id`=15;

/\* 25 Update address of customer with account no=1233108100 \*/

UPDATE `CUSTOMERS`

SET `Address`='Pune'

WHERE `Acc\_no`=1233108100;

/\* 26 Update the contact no of customer with Account no=1233108100 \*/

UPDATE `CUSTOMERS`

SET `Contact\_No`=8451864318

WHERE `Acc\_no`=1233108100;

/\* 27 retrieve all customer \*/

SELECT \*

FROM `CUSTOMERS`;

/\* 28 retrieve all the employee \*/

SELECT \*

FROM `Employee`;

/\* 29 Make sure Account Balance of cutomer is not NULl \*/

SELECT \*

FROM `CUSTOMERS`

WHERE `Amount` IS NOT NULL;

/\* 30 Update date of birth of customer with account no =1233108100 \*/

UPDATE `CUSTOMERS`

SET `DOB`='1999-5-27'

WHERE `Acc\_no`=1233108100;

/\* 31 Delete accout of cutomer which is not operated in 1 year \*/

/\* 32 Cancel the transaction with transaction id=17 \*/

DELETE FROM `Transaction`

WHERE `Trans\_id`=17;

/\* 33 Select all trancation with remark=Transfer \*/

SELECT \*

FROM `Transaction`

WHERE `Remark`='Transfer';

/\* 34 Select all transaction between account no=123310810 and account no=1239799042 \*/

SELECT \*

FROM `Transaction`

WHERE (`Acc\_no1`=1239799042 AND `Acc\_no2`=1239799042) OR (`Acc\_no1`=1239799042 AND `Acc\_no2`=123310810);

/\* 35 Erase all the transaction of account no=123310810 \*/

DELETE FROM `Transaction`

WHERE `Acc\_no1`=123310810 OR `Acc\_no2`=123310810;

/\* 36 Return the count of all customers in bank \*/

SELECT COUNT(`Acc\_no`)

FROM `CUSTOMERS`;

/\* 37 Return all the money in bank \*/

SELECT SUM(`Amount`)

FROM `CUSTOMERS`;

/\* 38 Return count of total employee in bank \*/

SELECT COUNT(`Emp\_id`)

FROM `Employee`;

/\* 39 Retrieve all customers with account balance greater than 100000 \*/

SELECT \*

FROM `CUSTOMERS`

WHERE `Amount`>100000;

/\* 40 Retrieve all customers with account balance less than 100000 \*/

SELECT \*

FROM `CUSTOMERS`

WHERE `Amount`<100000;

/\* 41 Select all the customers with account balance between 1000 4999 \*/

SELECT \*

FROM `CUSTOMERS`

WHERE `Amount` BETWEEN 1000 AND 4999;

/\* 42 Select all transaction with remark=Debit\*/

SELECT \*

FROM `Transaction`

WHERE `Remark`='Debit';

/\* 43 Delete all transactio with unsucessful remark \*/

DELETE FROM `Transaction`

WHERE `Remark`='unsucessful';

/\* 44 Select all customers who have done transaction today \*/

SELECT \*

FROM `CUSTOMERS`,`Transaction`

WHERE `Remark` LIKE =' ';

/\* 45 Select cutomers with atleast 10 number of transaction \*/

SELECT \* FROM `CUSTOMERS`

WHERE COUNT(`Trans\_id`)>=10;

/\* 46 Select customers with atmost 10 no of Transaction \*/

SELECT \* FROM `CUSTOMERS`

WHERE COUNT(`Trans\_id`)<=10

/\* 47 Selct all customers with no last name \*/

SELECT \*

FROM `CUSTOMERS`

WHERE `Last\_name` IS NULL ;

/\*48 Update minor account of customer \*/

UPDATE `CUSTOMERS`

SET `DOB`='2019-04-22'

WHERE `DOB`='2001-04-22';

/\* 49 Retrieve all transaction that took place today \*/

SELECT \*

FROM `Transaction`

WHERE `Date`='2019-04-20';

/\* 50 Return all cutomers with incomplete profile \*/

SELECT \*

FROM `Transaction`

WHERE `Contact\_No` IS NULL ;

/\* 51 Update Account Balance of customer with age greater than 60 \*/

UPDATE `CUSTOMERS`

SET `Amount`=`Amount`+1000;

WHERE `DOB` <='1959-04-20';

/\* 52 Update the account balance of customer every month \*/

UPDATE `CUSTOMERS`

SET `Amount`=`Amount`+100;

/\* 53 Select all transaction which are handled by emp\_id ATM\_BANKER \*/

SELECT \*

FROM `Transaction`

WHERE `Emp\_id`='ATM\_BANKER';

/\* 54 Update the phone no of cutomer with account no=1238069768 \*/

UPDATE `CUSTOMERS`

SET `Contact\_No`=9643158854

WHERE `Acc\_no`=1238069768;

/\* 55 Credit the account balance of customer with account no=1238069768 \*/

UPDATE `CUSTOMERS`

SET `Amount`=`Amount`+50000

WHERE `Acc\_no`=1238069768;

/\* 56 Debit account balance of customer with account no=1238069768 \*/

UPDATE `CUSTOMERS`

SET `Amount`=`Amount`-1000;

WHERE `Acc\_no`=1238069768;

/\* 57 Select all cutomers with bounced cheque \*/

SELECT \*

FROM `Transaction`

WHERE `Remark`='Bounced';

/\* 58 fine the customer with bounced cheque \*/

UPDATE `CUSTOMERS`

SET `Amount`=`Amount`-100;

WHERE `CUSTOMERS`.`Remark`='Bounced';

/\* 59 Count the no of sucessful transaction \*/

SELECT COUNT(`Trans\_id`)

FROM `Transaction`

WHERE `Remark`='unsucessful';

/\* 60 count no of suceesful transaction \*/

SELECT COUNT(`Trans\_id`)

FROM `Transaction`

WHERE `Remark`='Transfer' OR 'Debit' OR 'CREDIT\_CASH' OR 'CREDIT\_CHEQUE';

/\*61. Get the inner join of transaction made today and customer \*/

SELECT Date

FROM `transaction`

INNER JOIN Customer

ON `transaction.Date` = `Customer.Account\_No`;

/\* 63. Get the left join of transaction made today and customer \*/

SELECT Date

FROM `transaction`

LEFT JOIN `Customer`

ON `transaction.Date` = `Customer.Account\_No`;

/\* 64. Get the right join of transaction made today and customer \*/

SELECT Date

FROM `transaction`

RIGHT JOIN `Customer`

ON `transaction.Date` = `Customer.Account\_No`;

/\* 65. Get the count of number of employee with null password \*/

SELECT COUNT(`Password`)

FROM `Employee`

WHERE `Password`='NULL';

/\* 66. Get the count of number of customer with null bank balance \*/

SELECT COUNT(`Balance\_Amount`)

FROM `Customer`

WHERE `Balance\_Amount`='NULL';

/\* 67. Get customer with maximum number of transaction \*/

SELECT MAX(`Account\_No`)

FROM `Customer`

WHERE `transaction`='MAX';

/\* 68. Get customer with minimum number of transaction \*/

SELECT MIN(`Account\_No`)

FROM `Customer`

WHERE `transaction`='MIN';

/\* 69. Update account balance of each customer with 1000 having account balance greater than 100000 \*/

UPDATE `Customer`

SET `Account\_Balance` = '1000'

WHERE `Account\_Balance`>='100000';

/\* 70. Get name of customer with Account Balance greater than 100000 \*/

SELECT `Name`

FROM `Customer`

WHERE `Account\_Balance`>='100000';

/\* 71. Get name of customer with Account Balance lesser than 100 \*/

SELECT `Name`

FROM `Customer`

WHERE `Account\_Balance`<='100';

/\* 72. Get the oldest customer \*/

SELECT MAX(`Age`)

FROM `Customer`;

/\* 73. Get the youngest customer \*/

SELECT MIN(`Age`)

FROM `Customer`;

/\* 74. Get the count of successful transaction that took place today \*/

SELECT COUNT(`DATE`)

FROM `Transaction`

WHERE `Remark`='Successful';

/\* 75. Get the count of unsuccessful transaction that took place today \*/

SELECT COUNT(`DATE`)

FROM `Transaction`

WHERE `Remark`='Unsuccessful';

/\* 76. Count all customers below age 18 \*/

SELECT COUNT(`AGE`)

FROM `CUSTOMER`

WHERE `AGE`<'18';

/\* 77. Count all customers between age 18-60 \*/

SELECT `AGE`

FROM `CUSTOMER`

WHERE `AGE` BETWEEN '18' AND '60';

/\* 78. Count all customers above age 60 \*/

SELECT COUNT(`AGE`)

FROM `CUSTOMER`

WHERE `AGE`>='60';

/\* 79. Get the full join of transaction made today and customer \*/

SELECT `Date`

FROM `transaction`

FULL JOIN `Customer`

ON `transaction.Date` = `Customer.Account\_No`;

/\* 80. Return all the customers with null address \*/

SELECT `Name`

FROM `Customer`

WHERE `Address`='NULL';

/\* 81. Get the left join of number of transaction and customers \*/

SELECT `Date`

FROM `transaction`

LEFT JOIN `Customer`

ON `transaction.Date` = `Customer.Account\_No`;

/\* 82. Join the transaction done in last week and the customer \*/

SELECT `Date`

FROM `transaction`

JOIN `Customer`

ON `transaction.Date` = `Customer.Account\_No`;

/\* 83. Sort remarks with date \*/

SELECT `REMARKS`

FROM `TRANSACTION`

ORDER BY `Date`;

/\* 84. Sort the customers according to their age \*/

SELECT `Account\_No`

FROM `CUSTOMER`

ORDER BY `AGE`;

/\* 85. Sort the employees according to employee id \*/

SELECT `Employee\_id`

FROM `EMPLOYEE`

ORDER BY `Employee\_id`;

/\*86. Sort the customer according to their name \*/

SELECT `Account\_No`

FROM `Customer`

ORDER BY `Last\_Name`;

/\* 87. Sort the customers in descending order according to their age \*/

SELECT `Account\_No`

FROM `Customer`

ORDER BY `AGE` DESC;

/\* 88. Select all the successfull remark \*/

SELECT `REMARKS`

FROM `Transaction`

WHERE `REMARK`='Successful';

/\* 89. Select all the transaction \*/

SELECT \*

FROM `TRANSACTION`

/\* 90. Union all the customers with transaction id \*/

SELECT `Date` FROM `Transaction`

UNION

SELECT `Account\_No` FROM `Customer`;

/\* 91. Union all the employees with employee id \*/

SELECT `employee` FROM `Employee`

UNION ALL

SELECT `employee\_id` FROM `Employee`;

/\* 92. Count the number of bank account opened today \*/

SELECT COUNT(`Account\_No`)

FROM `Customer`

WHERE `Created\_On`='TODAY';

/\* 93. Sort the Customers according to their address \*/

SELECT `Account\_No`

FROM `Customer`

ORDER BY `ADDRESS`;

/\* 94. Return frequency of transaction done today \*/

SELECT COUNT(`Transaction\_Id`)

FROM `Transaction`

WHERE `Date`='Today';

/\* 95. Return the customers who have done transaction greater than 100000 \*/

SELECT COUNT(`Transaction\_Id`)

FROM `Transaction`

WHERE `Amount`>=100000;

/\* 96. Select all the customers without last name \*/

SELECT \* FROM `CUSTOMER`

WHERE `Last\_Name`='NULL';

/\* 97. Select all the customers without Email address \*/

SELECT \* FROM `CUSTOMER`

WHERE `E-mail`='NULL';

/\* 98. Sort the transaction done in descending order according to the amount \*/

SELECT `Transaction\_Id`

FROM `Transaction`

ORDER BY `Amount` DESC;

/\* 99. Select the customers having minor bank account \*/

SELECT \* FROM `CUSTOMERS`

WHERE `AGE`<'18';

/\* 100. Count the customers having minor bank account \*/

SELECT COUNT(`Account\_No`)

FROM `Customer`

WHERE `AGE`<'18';

/\* 101.Create a view having first\_name,last\_name,remark for customer\*/

CREATE VIEW customer\_remark AS

SELECT c.first\_name,c.last\_name,f.remark

FROM customer c

JOIN remark f ON f.customer\_id=c.customer\_id;

/\*102.Create trigger when a new customer is added\*/

CREATE TRIGGER inserted

AFTER UPDATE

INSERT INTO customer;

/\*103.Create assertion to check that all customers have first name\*/

CREATE ASSERTION first\_name\_exists

CHECK(NOT EXISTS(SELECT\*FROM CUSTOMER WHERE first\_name is NULL));