```sql
# Acquire the data required to build predictive model

# Insert a new column to complains table to show number of days required to resolve an issue


# add column

alter table complaints

add Days_to_complete decimal(10,0);

# input values to the column

Update complaints

SET Days_to_complete = CompletionDate - ComplainDate;

# obtain data without missing data

Select *

From complaints

Where CompletionDate != "NULL";


Create view No_Null AS

Select *

From complaints

Where CompletionDate != "NULL";

# create a subset

Select co.ID, concat_ws( " ", c.FirstName, c.LastName) as CustomerName,

    c.Gender, co.Days_to_complete as ResolveDays, co.ComplainStatusID as Status,

    co.ComplainPriorityID as Priority, co.ComplainTypeID as ComplainType,
p.ProductCategory,

    r.state_code as State, co.ClientSatisfaction

From No_Null as co, customers c, products p, regions r

Where co.CustomerID = c.CustomerID

AND co.ProductID = p.ProductID

AND c.RegionID = r.id;
```

```sql
# subset of data
Select co.ID, concat_ws( " ", c.FirstName, c.LastName) as CustomerName,
    c.Gender, r.state_code as State, p.ProductCategory, co.Days_to_complete as
ResolveDays,
    co.ComplainPriorityID as ResolvePriority, co.ComplainStatusID as ComplainStatus,
    co.ComplainTypeID as ComplainType,
    co.ClientSatisfaction
From complaints as co, customers c, products p, regions r
Where co.CustomerID = c.CustomerID
AND co.ProductID = p.ProductID
AND c.RegionID = r.id
AND co.CompletionDate != "NULL"
AND co.ClientSatisfaction != "N/A";

# create a view
Create View pred_model as
SELECT ID, Days_to_complete as ResolveDays, ComplainPriorityID, ComplainTypeID,
 ComplainCategoryID, ComplainStatusID, ClientSatisfaction, ExpectedReimbursement
FROM complaints
WHERE CompletionDate != "NULL"
AND ClientSatisfaction != "N/A";

Select *
From pred_model;

# update client satisfaction to a binary column
Update complaints
Set ClientSatisfaction = 1
Where ClientSatisfaction = "SAT";

Update complaints
Set ClientSatisfaction = 0
Where ClientSatisfaction = "NSA";
```

```sql
# subset query
SELECT ID, Days_to_complete as ResolveDays, ComplainPriorityID, ComplainTypeID,
 ComplainCategoryID, ComplainStatusID, ExpectedReimbursement, ClientSatisfaction
FROM complaints
WHERE CompletionDate != "NULL"
AND ClientSatisfaction != "N/A";

# get number of complains for each state
Select count(id), state_code
FROM regions
Group by state_code;

# create view
Create view reg_stat as
Select id, state_code
From regions;

# update regions
# aternate: use IF /WHEN statements to.
Update regions
set state_code = 1
where state_code = "AL";

Update regions
set state_code = 2
where state_code = "AR";

Update regions
set state_code = 3
where state_code = "AZ";

Update regions
set state_code = 4
where state_code = "CA";

Update regions
set state_code = 5
```

```sql
where state_code = "CO";

Update regions
set state_code = 6
where state_code = "CT";

Update regions
set state_code = 7
where state_code = "DC";

Update regions
set state_code = 8
where state_code = "DE";

Update regions
set state_code = 9
where state_code = "FL";

Update regions
set state_code = 10
where state_code = "GA";

Update regions
set state_code = 11
where state_code = "HI";

Update regions
set state_code = 12
where state_code = "IA";

Update regions
set state_code = 13
where state_code = "ID";

Update regions
set state_code = 14
where state_code = "IL";

Update regions
set state_code = 15
```

```sql
where state_code = "IN";

Update regions

set state_code = 16

where state_code = "KS";

Update regions

set state_code = 17

where state_code = "KY";

Update regions

set state_code = 18

where state_code = "LA";

Update regions

set state_code = 19

where state_code = "MA";

Update regions

set state_code = 20

where state_code = "MD";

Update regions

set state_code = 21

where state_code = "ME";

Update regions

set state_code = 22

where state_code = "MI";

Update regions

set state_code = 23

where state_code = "MN";

Update regions

set state_code = 24

where state_code = "MO";

Update regions

set state_code = 25
```

```sql
where state_code = "MS";

Update regions

set state_code = 26

where state_code = "MT";

Update regions

set state_code = 27

where state_code = "NC";

Update regions

set state_code = 28

where state_code = "ND";

Update regions

set state_code = 29

where state_code = "NE";

Update regions

set state_code = 30

where state_code = "NH";

Update regions

set state_code = 31

where state_code = "NJ";

Update regions

set state_code = 32

where state_code = "NM";

Update regions

set state_code = 33

where state_code = "NV";

Update regions

set state_code = 34

where state_code = "NY";

Update regions

set state_code = 35
```

where state_code = "OH";

Update regions

set state_code = 36

where state_code = "OK";

Update regions

set state_code = 37

where state_code = "OR";

Update regions

set state_code = 38

where state_code = "PA";

Update regions

set state_code = 39

where state_code = "RI";

Update regions

set state_code = 40

where state_code = "SC";

Update regions

set state_code = 41

where state_code = "SD";

Update regions

set state_code = 42

where state_code = "TN";

Update regions

set state_code = 43

where state_code = "TX";

Update regions

set state_code = 44

where state_code = "UT";

Update regions

set state_code = 45

```sql
where state_code = "VA";

Update regions

set state_code = 46

where state_code = "WA";

Update regions

set state_code = 47

where state_code = "WI";

Update regions

set state_code = 48

where state_code = "WY";
```

query to subset data

```sql
SELECT c.ID, r.state_code as StateID, c.Days_to_complete as ResolveDays, c.ComplainPriorityID, c.ComplainTypeID,

 c.ComplainCategoryID, c.ComplainStatusID, c.ExpectedReimbursement, c.ClientSatisfaction

FROM complaints c, regions r, customers cu

WHERE CompletionDate != "NULL"

AND ClientSatisfaction != "N/A"

ANd r.id = cu.RegionID

AND c.CustomerID = cu.CustomerID;
```

create binary for gender column

```sql
SELECT Gender,

CASE WHEN Gender = "FEMALE" THEN 1

        WHEN Gender = "MALE"   THEN 0

END AS Gender_Binary

FROM customers;
```

query to subset data

```sql
SELECT c.ID, r.state_code as StateID, s.Column1 as State, p.ProductCategory, p.Product, c.Days_to_complete as ResolveDays, c.ComplainPriorityID, c.ComplainTypeID,

 c.ComplainCategoryID, c.ComplainStatusID, c.ExpectedReimbursement, cu.Gender

FROM complaints c, regions r, customers cu, states s, products p
```

```sql
WHERE CompletionDate != "NULL"

AND ClientSatisfaction != "N/A"

AND Gender != ""

AND r.id = cu.RegionID

AND c.CustomerID = cu.CustomerID

AND s.Column2 = r.state

AND c.ProductID = p.ProductID;
```

query to subset data

```sql
SELECT c.ID, r.state_code as StateID, c.Days_to_complete as ResolveDays,
c.ComplainPriorityID, c.ComplainTypeID,

 c.ComplainCategoryID, c.ComplainStatusID, c.ExpectedReimbursement,
c.ComplainSourceID, c.ClientSatisfaction

FROM complaints c, regions r, customers cu

WHERE CompletionDate != "NULL"

AND ClientSatisfaction != "N/A"

ANd r.id = cu.RegionID

AND c.CustomerID = cu.CustomerID;
```

get the count of client satisfaction for each state

```sql
Select count(ClientSatisfaction), StateID

From cs_data

Group by StateID

order by StateID ASC;
```

update products

```sql
update products

set ProductCategory = 1

Where ProductCategory = "Business";


update products

set ProductCategory = 2

Where ProductCategory = "Capital Accumulation";
```

```sql
update products
set ProductCategory = 3
Where ProductCategory = "Family Protection";


update products
set ProductCategory = 4
Where ProductCategory = "Health";


update products
set ProductCategory = 5
Where ProductCategory = "Life";


update products
set ProductCategory = 6
Where ProductCategory = "Mortgaged";


update products
set ProductCategory = 7
Where ProductCategory = "Motor";


update products
set ProductCategory = 8
Where ProductCategory = "Other";


update products
set ProductCategory = 9
Where ProductCategory = "Property";
```